THE UNIVERSITY OF WESTERN ONTARIO
LONDON, CANADA

Computer Science 1026a

DEFERRED MIDTERM EXAMINATION: ANSWERS

November 1st

120 minutes

Name:_____

Student Number:_____

Lecture Section (circle one): T-Th 11:30 (MC) T-Th 3:30 (AHB)

Instructions (PLEASE READ):

- Fill in your name and student number above immediately.
- Have your student card out of its case and on the desk.
- For multiple choice and true/false questions, please circle the correct response on **this** exam paper.
- For short-answer questions, provide your answer in the space provided.
- This is a closed book exam
- If you finish within 10 minutes of the end of the exam, you must wait until the exam ends before leaving so as not to distract those who are still working.
- There is a blank page at the back of this exam for rough work. Additional sheets can be provided upon request. All paper must be returned to your instructor along with your exam.
- No electronic devices are allowed.
- **Please turn off your cell phone.**
- **DO NOT TURN THIS PAGE UNTIL DIRECTED TO DO SO**

Question	Out of	Mark
1. True/False	15	
2. Multiple Choice	15	
3. Short Answers	30	
4. Logic Errors	14	
5. A Little Code	26	
Total	100	

Question 1: True/False – 15 Marks (1 each)

For the following questions, please circle (or indicate as specified by the question) your answer directly on the exam sheet. Note that questions are each worth one point unless otherwise indicated.

1)	A variable in Python has a name and a location (memory).	**True**	False
2)	Python can have variables that are integers.	**True**	False
3)	Boolean variables can only have a value of true or false.	**True**	False
4)	Compilers translate source code into byte code	**True**	False
5)	The first position in a string in Python has the index 0.	**True**	False
6)	The symbol '#' is used in Python to indicate a comment.	**True**	False
7)	x1yZ is a valid variable name in Python.	**True**	False
8)	9.7E05 is a floating number in Python	**True**	False
9)	The keyword **def** is used to define a function in Python	**True**	False
10)	The assignment operator in Python is **=**	**True**	False
11)	To divide two integers to get an integer result, you can use **//**	**True**	False
12)	The keyword **elseif** can be used in if-statements in Python.	True	**False**
13)	The operator **+** can be used to concatenate two strings together	**True**	False
14)	In Python, lists are immutable	True	**False**
15)	`45 % 8` produces the result `5.0`	True	**False**

Question 2: Multiple Choice – 15 Marks (1 each)

1) Which of the following statements is/are TRUE about the CPU?

a. CPU stands for Central Processing Unit
b. The CPU is what performs computation
c. The CPU processes machine language
d. <u>At least two of the above statements are true</u>
e. None of the above are true

2) What are two of the most important benefits of the Python language?

a. Advanced mathematical equations and fast programs
b. Ease of use and fast programs
c. Ease of use and portability
d. Fast programs and smaller programs

3) Which statement(s) allows us to initialize the list `numbers` with 10 elements all set to zero?

a. `numbers = [0]`
b. `numbers[10] = 0`
c. `numbers = [0] * 10`
d. `numbers[10] = [0] * 10`

4) Which of the following subtracts a variable x from a variable y, divides their difference by 3 and adds 11 to the result:

a. ((x - y) / 3) + 11
b. x - y / 3 + 11
c. y - x /3 + 11
d. <u>(y - x) / 3 + 11</u>
e. None of the above are true

5) What will be the values of the variables num1 and num2 after the execution of the following assignments?

```
num1 = 21
num2 = 18
num1 = num1 + num2 / 2
num2 = num1
```

a. num1 is 21, num2 is 21
b. <u>num1 is 30, num2 is 30</u>
c. num1 is 30, num2 is 21
d. num1 is 30, num2 is 18
e. None of the above.

Consider the following code for the next two questions:

```
num1 = int(input("Enter a number: "))
num2 = int(input("Enter a number: "))
num3 = int(input("Enter a number: "))
if num1 > num2 :
  if num1 > num3 :
    print(num1)
  else :
    print(num3)
else :
  if num2 <= num3 :
    print(num2)
  else :
    print(num3)
```

6) Assuming that a user enters 30, 20, and 10 as the input values in that order to the code above, what is the output?

a. <u>30</u>
b. 20
c. 10
d. 30 and 20
e. Nothing, there is an error.

7) If the user enters, 10, 20, 30 as the input to the code above in that order, what is the output?

a. 10
b. <u>20</u>
c. 30
d. 10 and 30
e. Nothing, there is an error.

8) The following code snippet contains an error. What is the error?

```
cost = int(input("Enter the cost: "))
if cost > 100
   cost = cost - 10
print("Discounted cost:", cost)
```

a. Logical error: use of an uninitialized variable
b. <u>Syntax error: missing colon after if statement</u>
c. Syntax error: missing an else statement

d. Logical error: error in converting input

9) Which function call correctly invokes the partial drawShape function listed below and prints a star triangle?

```
def drawShape(type) :
   length = len(type)
   if length == 0 :
      return
   if type == "triangle" :
      print("  *")
      print(" ***")
      print("*****")
```

a. `drawShape(triangle)`
b. `drawShape("triangle")`
c. `drawShape`
d. `value = drawShape(triangle)`

10) Which of the following `for loops` will run the loop body 5 times?

a. `for i in range(13, 9, -1) :`
b. `for i in range(14, 10, -1) :`
c. `for i in range(15, 9, -1) :`
d. `for i in range(14, 9, -1) :`

11) Which of the following checks to see if there is a comma anywhere in the string variable `name`?

a. `if name.contains(",") :`
b. `if "," not in name :`
c. `if name.startswith(",") :`
d. `if "," in name :`

12) Is the code snippet written below legal?

```
s = "1234"
for i in range (0, 4) :
   print(s[i], s[i + 1])
```

a. Yes.
b. No; there should not be a colon at the end of line 2.
c. No; `for i = 3, s[i + 1]` will result in an string index out of range error.
d. No; `for i = 0, s[i]` will result in an string index out of range error

13) Which of the following statements is true about functions and strings:

a. A function can be called with a string as an argument.
b. A function can return a string.
c. Only a. is true.
d. Only b. is true.
e. <u>Both a. and b. are true.</u>

14) What does the following code snippet output?

```
a= 7
b= 8
def fun(b,a):
  a=9
  b=8
  return a

fun(a,b)
print(a,b)
```

a. 7 9
b. 8 9
c. 9 9
d. 8 8
e. <u>None of the above</u>

15)Which statement correctly creates a list that contains four elements?

a. `values[4]`
b. `values = [4]`
c. `values = [1, 2, 3, 4]`
d. `value[4] = [1, 2, 3, 4]`

Question 3: Short Answers – 30 Marks (2 each)

1) What is the value of `names` after the following code segment has run?

```
names = []
names.append("Amy")
names.append("Bob")
names.pop()
names.append("Peg")
names[0] = "Cy"
names.insert(0, "Ravi")
names.insert(4, "Savannah")
```

['Ravi', 'Cy', 'Peg', 'Savannah']

2) What is the output of the code snippet given below?

```
s = "zyxwv"
length = len(s)
i = 1
while i <= length // 2 :
   print(s[i-1], s[length - i])
   i = i + 1
```

z v
y w

3) What is printed by the following code snippet?

```
name = "This is London Ontario"
name = name.lower()
name = name.replace("o", "#")
name.upper()
print(name)
```

this is l#nd#n #ntari#

4) What is printed to the screen when this loop executes?

```
for i in range(24, 3, -7) :
   print(i, end = "-")
```

24-17-10-

5) What is printed from the following code snippet?

```
prices = [[ 1.0, 3.50, 7.50 ],
   [ 10.0, 30.50, 70.50 ],
   [ 100.0, 300.50, 700.50 ],
   [ 1000.0, 3000.50, 7000.50 ]]
print(prices[1][2])
```

70.5

Use the following code for the next three questions (i.e. questions 6 – 8)

```
num1 = int(input("Enter a number: "))
num2 = int(input("Enter a number: "))
num3 = int(input("Enter a number: "))
if not (num1 > num2 and num1 >= num3) :
   print("First num is", num1)
elif not(num2 > num1 and num2 > num3) :
   if num3 % 10 == 0:
       print("The value is", num3)
    if num1 % 10 == 0:
        print("The value is", num1)
    else:
        print("The value is", num2)
elif not (num3 > num1 or num3 > num2) :
   print(num3)
```

6) Assuming a user enters 40, 45, and 4 as the input, what is the output of the above code snippet?

First num is 40

7) Assuming a user enters 50, 45, and 10 as the input, what is the output of the above code snippet?

The value is 10
The value is 50

8) Assuming a user enters 7, 7, and 7 as the input, what is the output of the following code snippet?

First num is 7

9) What is missing from this code snippet to find the longest value in the list?

```
colors = ["red", "purple", "blue", "green", "yellow", "light green"]
longest = colors[0]
for i in range(1, len(colors)) :

    _____________________
       longest = colors[i]
```

if len(colors[i]) > len(longest) :

10) The following function interest(prin,rate,nyears) computes the total interest of an initial investment, prin, compounded annually at rate for nyears. Insert the correct variables and parameters into the code to complete it.

```
def interest(prin,rate,nyears):
    bal = ________                          prin
    totalint = 0
    for i in range(nyears):
        intrst = bal * ________             rate
        totalint = totalint + intrst
        bal = bal + ________                intrst
    return  _________                       totalint
```

11) What is the output of the following code snippet.

```
def myCalculator(n):

  i = 10
  x = 4
  y = 2

  while i >= 0 and n >=5:
       y = y ** n
       x = n % 4 + y
       i = i-1
   return x

print( myCalculator(3) )
```

4

12) The following code segment makes use of `while` statement. Convert it into code that does the same thing but makes use of a `for` statement.

```
i = 0
j = 0
while i  < 125 :
   i = i + 2
   j = j + 1
   print(j)
```

for i in range(0,125,2) : -- 2 marks if correct
** j=j+1 – 1 mark for correct body of loop**

13) Review the code snippet below:

```
sentence = input("Enter some text: ")
firstCh = sentence[0]
print(firstCh)
```

Write an if statement that correctly determines if the last letter of the string contained in `sentence` is an uppercase letter?

if sentence[len(sentence)-1].isupper():

_____

14) Given the code snippet below, what is returned by the function call:
`mystery(mystery(2, 2), mystery(2, 2))`?

```
def mystery(num1, num2) :
   result = num1 * num2
   return result
```

_____ **16**

15) What code completes this code snippet to swap the first and last element in the list?

```
states = ["Alaska", "Hawaii", "Florida", "Maine"]
i = 0

_____________________
temp = states[j]
states[j] = states[0]
states[i] = temp
```

```
j = len(states) - 1
```

Question 4: Logic Errors - Correcting Code Segments – 14 Marks

1) The following code segment prompts the user for integers and computes and displays all the prime factors of an integer greater than 1. Some factors may be repeated. For instance, if a user enters the number 12 the factors that will be printed are 2, 2 and 3. The program has four logic errors. Identify them and correct the lines of code that contain them. Note: a line may contain more than one logic error. *(6 Marks)*

```
# Read an integer from the user.
value = input("Enter an integer: ")          #need to add  int 2pt

# Compute and display the factors.
print("The factors are:")
while value >= 1 :                           #should be > not >=
   divisor = 2
   found = False

   # Search for a factor (divisor) until we find one.
   while found == True :                     #should be False not True  2pt
      if value % divisor == 0 :
         print(divisor)
         value = value / divisor
         found = True
      divisor = divisor - 1                  #should be +1 not -1
print('Goodbye')
```

2) The following program should display to the screen the reverse of string in all capital letters or all small letters depending on the first letter of the string that is input. The program has a function myReverse that is supposed to return the reverse of the original string and should convert it to all capitals or all small letters. If the first letter of the original string is a capital letter then the result string should be all capitals, otherwise it should be all small letters. The code is syntactically correct, but contains logic errors. Identify them and correct the lines in which they occur. Note: a line may contain more than one logic error. *(8 Marks)*

```
def myReverse(str):
    outputStr = ""
    i = 1                                   i = 0   (1 pt)
    while i < len(str)  :
        outputStr = outputStr  + str[i]     outputStr = str[i] + outputStr   (1 pt)
        i = i + 1
    if str[0].isupper :                     if str[0].isupper()        (2 pt)
        outputStr = str.upper()             outputStr = outputStr.upper()   (1 pt)
    else:
        outputStr = str.lower()             outputStr = outputStr.lower()   (1 pt)
    return str                              return outputStr    (2 pt)

inp = input('Enter the string to be reversed:')
print(myReverse(inp))
```

Question 5: A Little Code - 26 Marks

1) (16 marks) Create a program that keeps on processing input until the user enters a string that has less than 3 characters in it.

Processing:

If the input has the special character $ in it, the $ should be replaced with the *.
If the input has an even number of characters, the string should be converted to lowercase.
If the input ends with a capital letter, the string should be converted to uppercase.

The program should group the strings such that strings that begin with a number are together and strings that do not begin with a number are together. After the sentinel condition is met, the program should print both groups.

If the user has input the following words: CHAR, 12000, 3456, house, tabe, boY, 100456, at
The output should be

Words beginning with numbers: 3456, 100456
Words that do not begin with numbers: char, *12000*, house, tabe, BOY

```
strInp = input('Please enter the string input: ')
listOfNumbers = []
listOfNonNumbers = []
while len(strInp) > 2:
    strInp = strInp.replace('$', '*')
    if len(strInp) % 2 ==0:
        strInp = strInp.lower()
    if strInp[len(strInp)-1].isupper():
        strInp = strInp.upper()
    if strInp[0].isdigit():
        listOfNumbers.append(strInp)
    else:
        listOfNonNumbers.append(strInp)
    strInp = input('Please enter the string input: ')
print("Words that begin with a number:", str(listOfNumbers).strip("[]"))
print("Words that do not begin with numbers:",
str(listOfNonNumbers).strip("[]"))
```

2) (10 marks) Write a program that has two functions.
The first function `processNumber(str1)` that takes in a string, `str1` and **returns** the sum of the digits that make up the string.
The second function `processString(str1)` takes in a string, and determines how many number of times the special character # exists in the string and **print this number** to the screen

Write a program that uses these two functions. If the input entered only contains numbers then the `processNumber` function should be called, if the input does not contain only numbers then the `processString` function should be called.

Example
If str1 is *"10231"*
Your program should output
The sum of all the numbers in 10231 is 7

If str1 is *"oh my word #sofunny but not sure what happened #confused as well"*
The character # appears 2 times in the string

```
def processNumber(str1):
    sumTotal = 0
    for el in str1:
        sumTotal = sumTotal +int(el)
    return sumTotal

def processString(str2):
    count = str2.count('#')
    print("The character # appears", count, 'times in the string')


inp = input('enter the string')
if inp.isdigit():
    print("The sum of all the numbers in", inp, 'is', processNumber(inp))
else:
    processString(inp)
```

Python string functions

Function Name	Description
capitalize()	Returns the String with first character capitalized and rest of the characters in lower case.
lower()	Converts all the characters of the String to lowercase.
upper()	Converts all the characters of the String to uppercase.
count(str[,beg,end]])	Returns the number of times substring 'str' occurs in range [beg, end] if beg and end index are given. If it is not given then substring is searched in the whole String. Search is case-sensitive.
islower()	Returns 'True' if all the characters in the String are in lowercase. If any one character is in uppercase it will return 'False'.
isupper()	Returns 'True' if all the characters in the String are in uppercase. If any one character is in lowercase it will return 'False'.
isdecimal()	Returns 'True' if all the characters in String are decimal. If anyone character in the String is of other data-type, it will return 'False'.
isdigit()	Returns 'True' for any character for which isdecimal() would return 'True and some characters in 'No' category. If there are any characters other than these, it will return 'False'.
isalpha()	Returns 'True' if String contains at least one character (non-empty String) and all the characters are alphabetic, 'False' otherwise.
isalnum()	Returns 'True' if String contains at least one character (non-empty String) and all the characters are either alphabetic or decimal digits, 'False' otherwise.
find(str [,i [,j]])	Searches for 'str' in complete String (if i and j not defined) or in a sub-string of String (if i and j are defined).This function returns the index if 'str' is found else returns '-1', where, i=search starts from this index. j=search ends at this index.
index(str[,i [,j]])	This is same as 'find' method. The only difference is that it raises 'ValueError' exception if 'str' is not found.
rfind(str[,i [,j]])	This is same as find() just that this function returns the last index where 'str' is found. If 'str' is not found it returns '-1'.
count(str[,i [,j]])	Returns the number of occurrences of substring 'str' in the String. Searches for 'str' in complete String (if i and j not defined) or in a sub-string of String (if i and j are defined), where, i=search starts from this index, j=search ends at this index.
replace(old,new[,count])	Replaces all the occurrences of substring 'old' with 'new' in the String. If 'count' is defined then only 'count' number of occurrences of 'old' will be replaced with 'new', where, old =substring to be replaced, new =substring that will replace the old, count =number of occurrences of old that will be replaced with new.
split([sep[,maxsplit]])	Returns a list of substring obtained after splitting the String with 'sep' as delimiter, where, sep= delimiter, default is space, maxsplit= number of splits to be done.
lstrip([chars])	Returns a String after removing the characters from the beginning of the String. where, Chars=this is the character to be trimmed from the String. Default is whitespace character.
rstrip()	Returns a String after removing the characters from the End of the String. where, Chars=this is the character to be trimmed from the String. Default is whitespace character.
len(string)	Returns the length of given String

This page is left blank for rough work.