COSC 101, Exam #1
15 Febuary 2016

Name: ______________________________ Section: 10:20 / 11:20

You have 50 minutes to complete this exam.

There are 5 questions and a total of 46 points available for this exam. Don't spend too much time on any one question.

Since indentation is important in Python, please be sure that your use of indentation is obvious for any code you write.

If you want partial credit, show as much of your work and thought process as possible.

If you run out of space for answering a question, you can continue your answer on one of the blank pages at the end of the exam. If you do so, be sure to indicate this in two places: (1) below the question, indicate which blank page contains your answer, and (2) on the blank page, indicate which question you are answering.

Question	Points	Score
1	8	
2	8	
3	10	
4	10	
5	10	
Total:	46	

1. (8 points) Assume that the following statements have already been executed:

```
day = 'monday'
sunny = True
tmp = -1.5
month = 2
zipc = "13346"
```

For each of the following expressions, evaluate the expression and write the resulting value, or identify the error in the code that would prevent it from running.

(a) `tmp / (month-2)`

Solution:

```
Error: float division by zero
```

(b) `month * str(month)`

Solution:

```
'22'
```

(c) `month + zipc`

Solution:

```
Error: unsupported operand type(s) for +: 'int' and 'str'
```

(d) `len(day) < 4 or max(3 ** 2, 13) > len(zipc)`

Solution:

```
True
```

(e) `day[0] + 2 * day[1] + day[-4]`

Solution:

```
'moon'
```

(f) `float ((month+4) / len(zipc))`

Solution:

```
1.0
```

(g) `zipc + " " + 10 + day`

Solution:

```
Error: cannot concatenate 'str' and 'int' objects
```

(h) `tmp < (month % 2) and int(zipc[1]) < month`

Solution:

```
False
```

2. (a) (4 points) What is the output of the following program?

```
people = 30
bikes = 28
cars = 10

if (bikes + cars > people):
    print "no need of walking..."
    if cars > bikes:
        print "too many cars"
    else:
        print "bikes may work"
if bikes > people:
    print "bikes are cool"
else:
    print "let’s stay home then"
```

Solution:

```
no need of walking...
bikes may work
```

(b) (4 points) What is the output of the following program?

```
size = 5
for i in range(size):
    print "!",
    for j in range(i):
        print "*",
    for j in range(1):
        print i + j
print size * " -"
```

Solution:

```
! 0
! * 1
! * * 2
! * * * 3
! * * * * 4
 - - - - -
```

3. (10 points) Write a program that asks a user for his/her team name and prints a “cheer” in a particular format. If the user types "Jays", the output should be:

```
Jays!
goJays!
gogoJays!
gogogoJays!
```

That is the number of lines the program prints is equal to the number of charaters in the team name and the number of times "go" is printed increases by one on each line. Each line should end with an exclamation point – *and there should be no space between the exclamation point and the cheer.*

As another example, if the user types the name "raiders", the output should be:

```
raiders!
goraiders!
gogoraiders!
gogogoraiders!
gogogogoraiders!
gogogogogoraiders!
gogogogogogoraiders!
```

Solution:

```
team = raw_input("enter your team ")
go = ''
for char in team:
    print go + team + "!"
    go = "go" + go
```

4. (10 points) Weather has been highly variant this winter, and most people are waiting for warmer weather. To determine whether things are getting warmer or colder, write a program that compares the average temperature of the last several days to the temperature of the last day. If the last day's temperature is warmer than the average, then it's getting warmer! We also keep track of which day was the coldest recently, so that we can tell everyone outside of Hamilton what we have been through.

```
How many days of temperature history?: 3
Day 0 temperature: 4.5
Day 1 temperature: 5.3
Day 2 temperature: 1.0
It's getting colder... The worst temperature recently was 1.0
```

And another example where temperatures are improving:

```
How many days of temperature history?: 5
Day 0 temperature: 1.0
Day 1 temperature: -1.7
Day 2 temperature: 2.3
Day 3 temperature: 5.0
Day 4 temperature: 4.0
It's getting warmer! The worst temperature recently was -1.7
```

Your program must match the format shown in the example below *exactly*. Note that in the last example, even though the last day is not as warm as the day before, it is still warmer than the average, so we still declare this to be a warming trend. You may assume that temperatures are measured in Fahrenheit, and will not be above the world temperature record of 136 degrees.

Solution:

```
numdays = int(raw_input("How many days of temperature history?: "))

total = 0.0
mintmp = 100.0

for day in range(numdays):
    tmp = float(raw_input("Day " + str(day) + " temperature: "))
    total += tmp
    if tmp < mintmp:
        mintmp = tmp

avgtmp = total / numdays

if avgtmp > tmp:
    print  "It’s getting colder...",
else:
    print  "It’s getting warmer!",

print "The worst temperature recently was", mintmp
```

5. (10 points) Write a program that asks the user to enter a phrase. Your program must find the starting index of the last set of repeated letters. Some examples are below:

   - If the user enters `'duuuplicates'`, your program should compute an index of 1.
   - If the user enters `'duuuplicaates'`, your program should compute an index of 8, because that is the index of the first of the pair of `'a'`s.
   - If the user enters `'no duplicates'`, your program should indicate that there are no sets of repeated letters.

   Your programs output should be formatted as shown in the examples below:

```
Enter a phrase: duuplicatess
Last set starts at index 10

Enter a phrase: a phrase
No duplicates
```

Solution:

```
phrase = raw_input("Enter a phrase: ")
lastStrike = -1
char = ''

for idx in range(len(phrase)-1):
    if phrase[idx] == phrase[idx+1]:
        if char != phrase[idx]:
            lastStrike = idx
    char = phrase[idx]

if lastStrike != -1:
    print "Last duplicate at index", lastStrike
else:
    print "No duplicates"
```

(This page is intentionally blank. Label any work with the corresponding problem number.)

(This page is intentionally blank. Label any work with the corresponding problem number.)