CS150 Fall 2018 – Midterm

Name: _____ **Section: A / B**

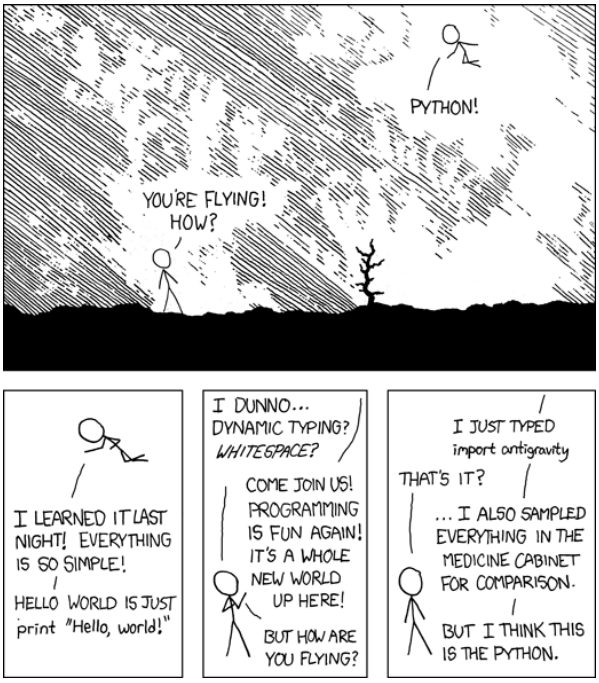
Date: _____ **Start time:** _____ **End time:** _____

Honor Code:

Signature: _____

This exam is closed book, closed notes, closed computer, closed calculator, etc. You may only use (1) the midterm "cheat sheet" from the course page, and (2) a single double-sided letter sheet of notes. **You have 2.5 hours.** Read the problem descriptions carefully and write your answers clearly and *legibly* in the space provided. Circle or otherwise indicate your answer if it might not be easily identified. You may use extra sheets of paper, stapled to your exam, if you need more room, as long as the problem number is clearly labeled and your name is on the paper. If you attached extra sheets indicate on your main exam paper to look for the extra sheets for that problem.

You do not need to include comments, docstrings or constants in your code.



1: Warming up	16	
2: Slice and dice	12	
3: Function calls	8	
4: T/F	8	
5: We've got problems	16	
6: Coding	16	
7: Turtle fun	10	
Total	86	

1. **[16 points]** Warming up

   (a) **[3 points]** Write out three (3) different ways of importing the `random` module and for each of those three ways, write an example of how you would invoke the `randint` function with that import style.

   (b) **[5 points]** Quick coding: In the game Yahtzee, a player can roll a variable number of six-sided dice (with numbers 1-6) in each turn. Write a function named `yahtzee` that takes the number of dice to roll as a parameter and returns a string with the randomly generated rolls for that number of dice. For example:

```
>>> yahtzee(1)
1
>>> yahtzee(2)
36
>>> yahtzee(5)
12345
```

(c) **[8 points]** Quick coding: Write two functions, one using a `for` loop and the other using a `while` loop, to print the numbers from 1 to 60 inclusive, one number per line, which are both even and evenly divisible by 3.

2. [**12 points**] Slice and dice

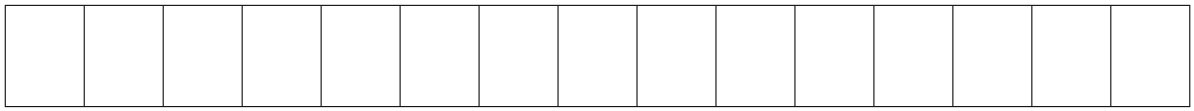
Given the variables `s` and `t` with the following values:

```
>>> s
'string games'
>>> t
'in cs class'
```

Evaluate the following expressions and provide the resulting value in the boxes, one character per box. Fill in any unused boxes at the end of the string. Make sure upper case letters can be clearly distinguished from lower case letters.

(a) `s[:6] + s[-1] + " " + t[:5]`

(b) `t[6::3] * 4`



(c) `t[3:6] + t.split()[0].upper() + t[5:]`

(d) `t[-s.find("i")] + s[6:11:1]`

3. [**8 points**] Function calls

Consider the following Python code:

```
def bar(r):
    print(r * 2)
    return int(r) * 2

def foo(s):
    r = 0
    for i in range(1, len(s)):
        r += bar(s[-i])
    return r

y = foo("4253")
```

After execution what is the value of `y` and what, if anything, is printed in the shell?

4. [**8 points**] T/F

For each of the statements below state whether they are **T**rue or **F**alse.

_____ `list("dcba") == ["a", "b", "c", "d"]` will evaluate to `True`

_____ `3/2 + 3//2 + 7 % 5` evaluates to `4.5`

_____ `(["abc"] + ["def"])[1][2]` will evaluate to `"b"`

_____ After this code executes `x` has the value 9

```
x = 0
for i in range(3):
    for j in range(3):
        x += i * j
```

_____ The following two functions produce identical results

```
def f(a, b):
    return a < 0 and b > 10

def g(a, b):
    return not (b <= 10 or a >= 0)
```

_____ The following function will execute successfully when the argument is a list of integers but will raise an error for a string argument

```
def mystery(x):
    for i in x:
        print(i * 2)
```

_____ All values in a list must be of the same type

_____ The following code will have an infinite loop for certain user inputs

```
i = int(input("Enter a number: "))
while i > 10:
    i -= 1
```

5. **[16 points]** We've got problems

(a) **[8 points]** The function below has two integer parameters, `a` and `b`. The function works as desired, however, it uses bad coding style. Rewrite the function to have identical behavior, but to be as concise as possible and implemented with good style.

```
def could_be_better(a, b):
    if a == 5 and a != 5:
        return True
    elif a == 5:
        return False
    elif b > 5 or False:
        return False
    else:
        return True
```

(b) **[8 points]** The following function was designed to return a list of indices at which the string "ATG" (the start codon) is found in the string parameter `dna`. There are several problems with this code including at least one syntax error, one runtime error and one logical error (the code executes but produces incorrect results). Identify and describe one syntax error, one runtime error and one logical error (for three errors total), which are not variations of the same issue. Your problems should impact correctness, not just style.

```
def starts(dna):
    for i range(len(dna)):
        if dna[i].lower() == "atg":
            indices = indices + [i]
    return indices
```

6. [**16 points**] Coding You are performing a linguistic study of tweets. You want to extract from a file of tweets just those tweets that include certain strings. Write a function named `filter_tweets` that has two parameters, a filename and a list of strings, and returns a list of tweets that include those strings (anywhere within the tweet). Assume the file contains one tweet per line. You should only return tweets that contain an exact match (including case) for one of the strings, i.e. if the string is "abc" the tweet "ABC" should not be returned but "def abc" should be returned. The returned list should only contain one instance of a tweet even if it matches multiple strings. You are **not** permitted to use a set or dictionary. Your code need not be a single function, you can write other functions to be invoked from `filter_tweets`.

```
>>> filter_tweets("tweets.txt", ["DM", "@Middlebury", "@Midd_Dining"])
["@Middlebury Class of 2018 celebrates ...", "Don't DM me if...", ...]
```

7. [**10 points**] Turtle fun

```
from turtle import *

def shape(x):
    forward(x)
    left(30)
    backward(x)

left(75)
side = 125
while side >= 50:
    shape(side)
    right(30)
    side = side - 25
```

Draw the image that would be created by the above code and label your drawing with relevant dimensions and angles, e.g. lengths of lines. Assume that the turtle is initially at the origin, facing right.

Page intentionally left blank.