

# First Midterm

You have 75 minutes (until the end of the class period) to complete this exam. There are 50 points possible, so allow approximately one minute per point and you'll have plenty of time left over.

Please read all the problems carefully. If you have a question on what a problem means or what it calls for, ask us. Unless a problem specifically asks about errors, you should assume that each problem is correct and solvable; ask us if you believe otherwise.

In answering these questions, you may use any Python 3 features we have covered in class, in the text, in the lab assignments, or earlier on the exam, unless a problem says otherwise. Use more advanced features at your own risk; you must use them correctly. If a question asks for a single item (e.g., one word, identifier, or constant), supplying more than one will probably not receive credit.

Remember, stay cool! If you run into trouble on a problem, go on to the next one. Later on, you can go back if you have time. Don't let yourself get stuck on any one problem.

You may not share any information or materials with classmates during the exam and you may not use any electronic devices, including cellphones.

Please write your answers clearly and neatly—we can't give you credit if we can't decipher what you've written.

We'll give partial credit for partially correct answers, so writing something is better than writing nothing. But be sure to answer just what the question asks.

Good luck!

**Problem 1**  
(12 points)

**Problem 2**  
(10 points)

**Problem 3**  
(10 points)

**Problem 4**  
(4 points)

**Problem 5**  
(4 points)

**Problem 6**  
(7 points)

**Problem 6**  
(3 points)

**Total**  
(50 points)

**Problem 1** (12 points) **Topic: Simple expressions**

Use the following definitions in this problem:

```
m = 25
x = 'turnip'
y = 'leek'
L = ['Ford', 'Carter', 'Reagan', 'Bush', 'Clinton', 'Bush', 'Obama']
```

**(a)** (5 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.)

```
print('Winter 2013')      Winter 2013 [credit with or without surrounding quotes] 0.5 pt
print("Let's go!")       Let's go! [credit with or without surrounding quotes, but Let's must be right] 0.5pt

print(5 * 6 == 30)       True 0.5pt
print((m * 4) - 3)       97 0.5pt
print(m / 5)             5.0 0.5pt [credit for just 5, too]
print(m + 3 >= 30)       False 0.5pt

print(x + '***' + y)     turnip***leek 1pt
print(len(x) + len(y))   10 1pt
```

**(b)** (4 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.) Remember zero-based indexing.

```
print(x[1])              # u 0.5pt
print(y[-1])            # k 0.5pt

print(L[0])             # Ford 0.5 pt
print(len(L))           # 7 0.5 pt

print('Pres.', L[-3])    # Pres. Clinton (credit for space or no space after period) 1pt
print('Mr. ' + L[3])     # Mr. Bush (-1/2 for quotes surrounding whole result; credit for Mr.Bush) 1pt
```

**(c)** (3 points) Each of the following is a boolean expression (its value is either True or False); for each expression, give its value.

```
len(L) != 5             # True. SCORING: 0.5pt each
'Nixon' in L            # False

'n' in x                # True
'x' in 'Nixon'          # True
L[0] >= 'Nixon'         # False

len(5 * 'ok') == 5 * len('ok') # True
```

**Problem 2** (10 points) **Topic: Namedtuples**

(a) (3 points) The Anteater Vegetable Stand represents each kind of vegetable it sells in a namedtuple called `Vegetable` that has four fields: the kind of vegetable, the name of the grower, the quantity they have in stock, and the price (per pound).

Which of the following defines a namedtuple that satisfies this specification? Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

- A. `Celery = Vegetable('celery ZotFarms 225 1.39')`
- B. `Vegetable = namedtuple('Vegetable', 'name farm instock price')` **# This one**
- C. `Vegetable = namedtuple('Vegetable', 'kind of vegetable, grower name, amt available, price per lb')`
- D. `Celery = namedtuple('Vegetable', 'celery', 'Zot Farms', 225, 1.39)`
- E. `Vegetable = namedtuple('Vegetable', 'variety grower quantity price')` **# This one**

(b) (3 points) Which of the following creates a `Vegetable` object as a namedtuple following the description above, to represent 125 pounds of celery grown by Zot Farms, on sale for \$2.50 per pound? Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

- A. `veg1 = Vegetable('celery Zotfarms 125 2.50')`
- B. `veg1 = Vegetable.celery('Zot Farms', 125, 2.50)`
- C. `veg1 = tomato.quantity * celery.price`
- D. `veg1 = Vegetable('celery', 'Zot Farms', 125, 2.50)` **# This one**
- E. `veg1 = Celery('Zot Farms', 125, 2.50)`

(c) (3 points) Which of the following statements reflects a price decrease of \$0.50 for the `Vegetable` object stored in `veg1`? [You may assume any of the correct `Vegetable` definitions above.] Circle each correct statement; the correct answer may include *one or more* of A, B, C, D, or E.

- A. `veg1 = Vegetable(veg1.name, veg1.farm, veg1.instock, veg1.price - 0.50)` **# This one**
- B. `veg1 = Vegetable('celery Zot Farms 125 -0.50')`
- C. `veg1 = veg1._replace(price = 2.00)` **# This one**
- D. `veg1.price = veg1.price - 0.50`
- E. `veg1 = veg1._replace(price = veg1.price - 0.50)` **# This one**

(d) (1 point) Of the possible solutions in part (c) above, one is technically correct but of lower quality than the others. Which solution is lower quality and why? Choose the *one* most correct, most complete answer; don't circle more than one.

- A. `veg1 = veg1._replace(price = veg1.price - 0.50)` because underscores are hard to type
- B. `veg1 = Vegetable(veg1.name, veg1.farm, veg1.instock, veg1.price - 0.50)` because it's requires more typing than the others.
- C. `veg1 = veg1._replace(price = 2.00)` because it's inflexible; it's only right if the original price is \$2.50 **# This one**
- D. `veg1.price = veg1.price - 0.50` because it's the shortest

**Problem 3** (10 points) **Topic: Functions, sorting a list of namedtuples**

Professor Andrea Anteater represents the students in her class with a namedtuple defined as follows:

```
Student = namedtuple('Student', 'name ID midterm project final')
```

where the name is a string, the ID is an int, and the remaining fields are floats storing the scores (in the range 0 to 100) on the indicated items in the class.

(a) (2 points) In the function definition below, fill in each blank with one Python variable name, function name, method name, constant, or operator to satisfy the problem specification.

```
def total_score(s: Student) -> float:
    ''' Return the student's score in the class, with the midterm worth 20%,
        the project 30%, and the final 50%
    '''
    return s.midterm * _____ + s._____ * 0.30 _____ s.final _____ 0.50
    return           0.20           project           +           * (SCORE 1/2 pt ea)
```

(b) (4 points) What do the following statements print? [The arithmetic is easy to do in your head.]

```
s1 = Student('Zot, Zoe', 11223344, 100, 100, 100)
s2 = Student('Irvine, Irving', 55667788, 50, 50, 50)
print(s1.name, total_score(s1))
print(s2.name, total_score(s2))
```

**Zot, Zoe 100.0**                      **SCORING: 1/2 point for each name, 1 point for each correct number with the right**  
**Irvine, Irving 50.0**                **name (omitting .0 okay), 1 point for ev. else correct (without worrying abt extra spacing)**

(c) (2 points) In the function definition below, fill in each blank with one Python variable name, function name, method name, constant, or operator to satisfy the problem specification.

```
def project_score(s: Student) -> float:
    ''' Return the student's score on the project
    '''
    return _____ . _____ s.project (SCORING 1 pt each)
```

(d) (2 points) In addition to the function `project_score` defined above, assume you have also defined similar functions called `midterm_score` and `final_score` that return the values of those fields.

If Prof. Anteater has a list of 300 students called `SL`, fill in the blanks below to reorder `SL` by each student's final exam score, lowest to highest.

```
_____.sort(key = _____)    SL    final_score
```

**Problem 4** (4 points) **Topic: For loops**

Suppose we have a list of Students called SL, as in the previous problem.

Match the four for-loops below (A through D) with the most accurate description below.

A. 

```
for s in SL:
    print(s.name)
```

B. 

```
for s in SL:
    print(s.name, total_score(s))
```

C. 

```
for s in SL:
    print(s, total_score(s))
```

D. 

```
for s in SL:
    print(s.name, s.total_score)
```

\_\_\_ Print the names of the students, one per line **...(s.name)...**

\_\_\_ Print the name of each student with his or her overall score in the class **...(s.name, total\_score(s))...**

\_\_\_ Produces an error message about the improper use of total\_score **...(s.name, s.total\_score)...**

\_\_\_ Produces the largest volume of output **...(s, total\_score(s))...**

**Problem 5** (4 points) **Topic: Identifying types**

Identify the data type of each of the following expressions, using definitions that appear in this exam where appropriate. Choose from:

int      float      bool      str      list      Vegetable      Student

A. SL **list**

B. veg1 **Vegetable**

C. SL[0] **Student**

D. veg1.price **float**

E. veg1.price > 10.00 **bool**

F. SL[1].name **str**

G. [3, 4, 5, 6, 7] **list**

H. len(SL[5].name) **int**

**Problem 6** (7 points) **Topic: Control flow with if and for**

(a) (3 points) What does the following code print out?

```
a = 5
for n in range(a):
    print(n, a)
print('MLK Day')
```

**0 5** **SCORING:****1 5** **1/2 point for five lines of two numbers each****2 5** **1/2 point for first column going 0 to 4****3 5** **1/2 point for second column being all 5s****4 5** **1/2 pt for MLK Day at end****MLK Day** **up to 1 pt for everything else correct**

(b) (4 points) What does the following code print out?

```
print('Presidents Day')
for p in L: # Use the definition of L from Problem 1
    if len(p) > 4:
        print(p)
print('Monday holiday')
```

**Presidents Day** **SCORING:****Carter** **1/2 pt for Presidents Day once at top****Reagan** **2 pts for correct names in correct order****Clinton** **1/2 pt for Monday holiday once at bottom****Obama** **up to 1 pt for everything else correct****Monday holiday****Problem 7** (3 points) **Topic: Control flow in functions**

What does the following code print?

```
def alpha (n: int, s: str) -> str:
    return beta(s) * n
```

```
def beta (x: str) -> str:
    return x + "$"
```

```
print("Valentine's Day")
print(alpha(3, 'PQ'))
print("Good-bye")
```

**Valentine's Day** **SCORING:** **1/2 pt for Valentine's Day at top and Good-bye at bottom (both)****PQ\$PQ\$PQ\$** **1 pt for at least one PQ\$****Good-bye** **1 pt for three copies****1/2 pt for everything else correct**