

1. 8:00a Prateek B. Swamy
2. 10:00 Prateek B. Swamy
3. 12:00 Rohit Malhotra
4. 2:00 Rohit Malhotra
5. 4:00 Roeland Singer-Heinze
6. 6:00 Roeland Singer-Heinze
7. 8:00p Akshat Amrish Patel
8. 8:00a Shibani Konchady
9. 10:00 Shibani Konchady
10. 12:00 Vignesh Raghunathan
11. 2:00 Vignesh Raghunathan
12. 4:00 Akshat Amrish Patel
13. 8:00a Kartic Saxena
14. 10:00 Kartic Saxena
15. 12:00 Archit Dey
16. 2:00 Archit Dey

# First Midterm

You have 75 minutes (until the end of the class period) to complete this exam. There are 65 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 with or receive from anyone besides the instructor or TAs any information or materials during the exam. You may not use any electronic devices.

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!

<b>Problem 1</b> (11 points)
<b>Problem 2</b> (9 points)6
<b>Problem 3</b> (19 points)
<b>Problem 4</b> (5 points)
<b>Problem 5</b> (6 points)
<b>Problem 6</b> (6 points)
<b>Problem 7</b> (9 points)
<b>Total</b> (65 points)

**Problem 1** (11 points)

Use the following definitions in this problem:

```
THIS_YEAR = 2015
a = "UCI's 50th"
L = ['class', 'book', 'computer', 'dorm', 'roommate', 'professor', 'smartphone']
```

(a) (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.)

```
print("Go Eaters!")
print(THIS_YEAR * 10)
print(THIS_YEAR, "is", a)
print(len(L) >= 10)
```

(b) (3 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(a[1])
print(a[0] == 'I')
print(a[2] + 'CE')
print('500' in a)
print(a in '50th')
print(len('OK'*3), 'OK'*3)
```

(c) (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(L[-1], L[0])
print(len(L), len(L[1:]))
print(L[1], L[3], L[5])
print("Dorm" in L)
```

**Problem 2** (9 points)

The Anteater Cafe represents each coffee drink in namedtuple called Drink that has four fields: the name of the drink, the size in ounces, customization notes (e.g., "decaf no-foam"), and its price.

(a) (3 points) Which of the following defines a namedtuple that satisfies this specification? Circle *one or more* of A, B, C, D, E, or F; more than one may be correct.

- A. `Coffee = namedtuple('Coffee', 'name ounces notes cost')`
- B. `Drink = namedtuple('Drink', 'Caffe Latte', 12, 'decaf no-foam', 3.55)`
- C. `Drink = namedtuple('Drink', 'Espresso, 3, none, 2.75')`
- D. `Drink = namedtuple('Drink', 'title ounces customization price')`
- E. `Drink = namedtuple('Drink', 'name size notes price')`
- F. `Drink = Drink('Cappucino', 20, 'half-caf nonfat', 4.45)`

**(b)** (3 points) Which of the following creates a new Drink object, following the definition above, to represent a 20-ounce iced decaf Americano that costs \$4.75? Circle *one or more* of A, B, C, D, or E; more than one may be correct.

- A. `new_drink = Drink.Americano(20, 'iced decaf', 4.75)`
- B. `new_drink = Drink('Americano', 20, 'iced decaf', 4.75)`
- C. `new_drink = Drink('Americano', 20, 'iced decaf', 4.75)`
- D. `new_drink = namedtuple('Drink', 'Americano 20 iced decaf 4.75')`
- E. `new_drink = Drink(20, 'iced decaf', 'Americano', 4.75)`

**(c)** (3 points) You want to increase the price of each drink by fifteen cents. Which of the following statements correctly increases the price of the Drink object stored in `a_drink` (according to at least one of the correct definitions in part (a))? Circle *one or more* of A, B, C, D, E, or F; more than one may be correct.

- A. `a_drink = a_drink._replace(price = price + 0.15)`
- B. `a_drink.price = a_drink.price + 0.15`
- C. `a_drink = Drink(a_drink.name, a_drink.size, a_drink.notes, a_drink.price + 0.15)`
- D. `a_drink = a_drink._replace(price = 4.90)`
- E. `a_drink = Drink(price = a_drink.price + 15)`
- F. `print(a_drink.price + 0.15)`

### Problem 3 (19 points)

The Anteater Cafe has branches worldwide. Their management has asked you to analyze a recent customer satisfaction survey. Customers at every branch have given the branch a 0–100 score in three categories: quality of the drinks, the service, and the ambience (physical environment). You use this namedtuple to represent a branch's survey results:

```
Branch = namedtuple('Branch', 'location ID drinks service ambience')
```

The location is a string, the ID is an int, and the remaining fields are floats storing the average rating from all customers on each category for that branch.

**(a)** (4 points) In the function definition below, fill in each blank with a single Python constant, operator, or identifier name (variable, function, attribute, method) to satisfy the problem specification.

```
def overall_satisfaction(b: Branch) -> float:
    """ Return the branch's overall satisfaction score, in the range 0-100, computed
        as an equally-weighted average of its drinks, service, and ambience scores.
    """
    return (_____.drinks _____ b.service + b._____) / _____
```

**(b)** (1 point) Fill in each blank in the assert statements below so that each assertion will be true (assuming a correct solution to part (a)).

```
assert overall_satisfaction(Branch('Irvine', 111, 25, 50, 75)) == _____
```

```
assert overall_satisfaction(Branch('Costa Mesa', 222, _____, _____, _____)) == 100
```

(c) (6 points) What do the following statements print? [You can do the arithmetic in your head.]

```
B1 = Branch('Anaheim', 333, 85, 90, 95)
B2 = Branch('Laguna Beach', 444, 93, 93, 93)
print(B1.location, "(#", B1.ID, "):", overall_satisfaction(B1), 'overall/100')
print(B2.location, "(#", B2.ID, "):", overall_satisfaction(B2), 'overall/100')
```

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

```
def Branch_drinks(b: Branch) -> float:
    ''' Return a Branch's score for drinks
    ...
    return _____ . _____
```

(e) (6 points) In addition to the function `Branch_drinks` defined above, assume you have also defined similar functions called `Branch_service` and `Branch_ambience` that return the values of those fields. Also, suppose that Anteater Cafe has a list of 750 Branches called `BL`

(e.1) Fill in the blanks below to reorder `BL` by each Branch's service score, lowest to highest:

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

(e.2) Fill in the blanks below to reorder `BL` by each Branch's overall satisfaction score, lowest to highest. You may use other functions that have been defined in this exam.

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

(e.3) Write a single Python statement similar to those in (e.1) and (e.2) to reorder `BL` by each Branch's overall satisfaction score, highest to lowest. You may use other functions that have been defined in this exam.

#### Problem 4 (5 points)

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

```
def service_higher (BL: 'list of Branch') -> 'list of Branch':
    """ Return a list of those Branches on the specified list whose service score is
        higher than its drinks score
    """
    result = [ ]
    for b in _____:
        if b._____ b.drinks:
            _____.append(b)
    return _____
```

**Problem 5** (6 points)

Suppose we have a list of Branches called `BL`, as in the previous problem. Fill in each blank with one of the ten code segments below (A through J) to match its most accurate description. Don't use any segment (A–J) more than once.

- A. 

```
for b in BL:
    print(b.location, b.overall_satisfaction)
```
- B. 

```
for b in BL:
    if b.ambience >= 75:
        print(b)
```
- C. 

```
for b in BL:
    if overall_satisfaction(b) > 90:
        print(b.location, b.ID)
```
- D. 

```
for b in BL:
    print(b.location, overall_satisfaction(b))
```
- E. 

```
for b in sorted(BL, key=overall_satisfaction, reverse=True):
    print(b.location, b.ID)
```
- F. 

```
for b in BL:
    if b.ambience >= 75:
        print(b.location)
```
- G. 

```
for b in BL:
    print(b)
```
- H. 

```
for b in BL:
    print(BL[b].location, overall_satisfaction(BL[b]))
```
- I. 

```
for b in BL:
    print(b.location, overall_satisfaction(b) > 90)
```
- J. 

```
for b in BL:
    print(b.location)
```

- \_\_\_ Produces an error message about the improper use of `overall_satisfaction`
- \_\_\_ Prints the location and ID of each Branch with overall satisfaction over 90
- \_\_\_ Prints the location of each Branch, one per line
- \_\_\_ Prints the location and overall satisfaction score for every Branch
- \_\_\_ Prints the locations of all Branches whose ambience is at least 75
- \_\_\_ Prints all the information about each Branch in Python namedtuple form

**Problem 6** (6 points)

Identify the data type of each of the following expressions, using definitions that appear in this exam.

Chose from: int float bool str Branch list of Branch list of str list of float function

'party' in L

overall\_satisfaction(B2)

a[0:3] + '!!'

L[3:]

len(L[4:])

2015

L[1]

B2

B2.drinks

[B2, B1]

a[1] == 'i'

Branch\_service

**Problem 7** (9 points)

What does the following code print out?

```
def chocolate (s: str) -> str:
    return "Choc." + s

def icing (n: int, w: str) -> str:
    return chocolate(w) * n

print("Here we go.")
print(chocolate("cake"))
print(icing(3, "blue"))
print("There we are.")
```

When you're done, please:

- Gather up all your stuff.
- Take your stuff and your exam down to the front of the room.
- Turn in your exam; show your ID if asked.
- Exit by the doors at the front of the room. Don't go back or disturb students still taking the test.