Week 4 Exercise

TOTAL POINTS 10

1. Consider this code: 1 point

```
class Contact:
    """ A contact with a first name, a last name, and an email address. """

    def __init__(self, first_name, last_name, email_address):
        """ (Contact, str, str, str) -> NoneType

        Initialize this Contact with first name first_name, last name
        last_name, and email address email_address.
        """

        self.first_name = first_name
        self.last_name = last_name
        self.email_address = email_address
```

Select the code fragment(s) that create and initialize a **Contact** using the constructor (method **__init__**).

- [x]
```
paul = Contact('Paul', 'Gries', 'paul@example.com')
```

- [ ]
```
contact = Contact()

paul = Contact(contact, 'Paul', 'Gries', 'paul@example.com')
```

- [ ]
```
paul = Contact()

paul.first_name = 'Paul'

paul.last_name = 'Gries'

paul.email_address = 'paul@example.com'
```

- [ ]
```
info = ['Paul', 'Gries', 'paul@example.com']

paul = Contact(info)
```

2. This question uses class **Contact** from the previous question. 1 point

Variable **jen** refers to a **Contact** object. Select the correct way to print **jen**'s email address.

- ( ) `print(jen.self.email_address)`
- (•) `print(jen.email_address)`
- ( ) `print(self.email_address)`
- ( ) `print(jen[2])`

3. This question uses class **Contact** from the previous questions. 1 point

Another method has been added to class **Contact**:

```
def add_phone_number(self, telephone_num):
    """ (Contact, str) -> NoneType

    Add phone number telephone_num for this contact.
    """

    self.phone_number = telephone_num
```

For a variable **khaled** that refers to a **Contact** object, which code fragment correctly calls method **add_phone_number**?

- (•) `khaled.add_phone_number('555-1111')`
- ( ) `khaled.add_phone_number() = '555-1111'`
- ( ) `khaled.add_phone_number(khaled, '555-1111')`
- ( ) `add_phone_number(khaled, '555-1111')`

4. This question uses class **Contact** from the previous questions, and also uses types **str**, **float**, and **list**. 1 point

Here are several code fragments. In each fragment, there is a pair of method calls. In some pairs, the two method calls are equivalent to each other, and in the others, the two method calls are not equivalent to each other. Select the code fragment(s) in which the method calls are equivalent to each other.

Assume that variable **c** refers to a **Contact** and that variable **L** refers to a **list**.

- [ ]
```
L.index(3)

list.index(3)
```

- [ ]
```
c.add_phone_number('555-1111')

c.add_phone_number(c1, '555-1111')
```

- [ ]
```
(0.6).as_integer_ratio()
```

```
3   float.as_integer_ratio(float, 0.6)
```

- [x]
```
1   str.replace('abc 123', '123', '246')
2   'abc 123'.replace('123', '246')
```

- [x]
```
1   c.add_phone_number('555-1111')
2
3   Contact.add_phone_number(c, '555-1111')
```

5. This question uses class **Contact** from the previous questions. 1 point

Variable **rorik** refers to a **Contact** object with instance variables **first_name**, **last_name** and **email_address** that refer to **'Rorik'**, **'Henrikson'** and **'rorik@example.com'** respectively.

What is produced when **str(rorik)** is called?

- (•) A string containing information about the object that **rorik** refers to. This string contains both its type and its memory address.
- ( ) **'Rorik Henrikson <rorik@example.com>'**
- ( ) **'Henrikson, Rorik <rorik@example.com>'**
- ( ) A string containing the types and memory addresses of the objects that **first_name**, **last_name**, and **email_address** refer to.

6. This question uses class **Contact** from the previous questions. 1 point

Another method has been added to class **Contact**:

```
def __str__(self):
    """ (Contact) -> str

    Return a string representation of this contact.
    """

    return '{0} {1} <{2}>'.format(self.first_name,
        self.last_name, self.email_address)
```

Variable **rorik** refers to a **Contact** object with instance variables **first_name**, **last_name**, and **email_address** that refer to **'Rorik'**, **'Henrikson'** and **'rorik@example.com'** respectively.

What is produced when **str(rorik)** is called?

- ( ) A string containing information about the object that **rorik** refers to. This string contains both its type and its memory address.
- (•) **'Rorik Henrikson <rorik@example.com>'**
- ( ) A string containing the types and memory addresses of the objects that **last_name**, **first_name**, and **email_address** refer to.
- ( ) **'Henrikson, Rorik <rorik@example.com>'**

7. This question uses class **Contact** from the previous questions. 1 point

Consider this code:

```
class Email:
    """ An email with a list of recipients, a subject and a body. """

    def __init__(self, recipients, subject, body):
        """ (Email, list of Contact, str, str) -> NoneType

        Initialize this Email with recipients, subject and body.       """

        self.recipients = recipients
        self.subject = subject
        self.body = body
```

Which of the following can be used to create an **Email** object?

- [x]
```
1   student1 = Contact('Hugh', 'Z.', 'hugh@fakedomain.com')
2
3   student2 = Contact('Kathryn', 'Z.', 'kathryn@fakedomain.com')
4
5   student3 = Contact('Karin', 'Z.', 'karin@fakedomain.com')
6
7   students = [student1, student2, student3]
8
9   subject = 'LTP2: E4 is posted!'
10
11  body = 'Hello,\nE4 is posted. Good luck!\n Paul and Jen'
12
13  new_email = Email(students, subject, body)
```

- [x]
```
1   :act('Kathryn', 'Z.', 'kathryn@fakedomain.com')], 'Hello', 'Hi there!\n Bye for now.')
```

- [ ]
```
1   new_email = Email()
```

- [ ]
```
1   new_email = Email('Hello', 'Hi there!\n Bye for now.')
```

8. This question uses classes **Contact** and **Email** from the previous questions. 1 point

This method is added to class **Email**:

```
1   def __str__(self):
```

```
def __str__(self):
    """ (Email) -> str

    Return a string representation of this email.
    """

    result = 'To: '
    for contact in self.recipients:
        result = result + '{0}, '.format(contact)

    result = result + '\nSubject: {0}'.format(self.subject)
    result = result + '\n{0}'.format(self.body)
    return result
```

Variable **message** refers to an **Email** object created with:

- recipients:
  [Contact('Paul', 'Gries', 'paul@example.com'), Contact('Jen', 'Campbell', 'jen@example.com')]
- subject: **'2nd MOOC'**, and
- body **'Hi!\nI hope your 2nd MOOC is going well!\nBye :-)'**.

What is printed when **print(message)** is executed?

- ( ) To: Contact('Paul', 'Gries', 'paul@example.com'), Contact('Jen', 'Campbell', 'jen@example.com')

  Subject: 2nd MOOC

  Hi!

  I hope your 2nd MOOC is going well!

  Bye :-)

- ( ) To: [Paul Gries <paul@example.com>, Jen Campbell <jen@example.com>]

  Subject: 2nd MOOC

  Hi!

  I hope your 2nd MOOC is going well!

  Bye :-)

- ( ) To: [Contact('Paul', 'Gries', 'paul@example.com'), Contact('Jen', 'Campbell', 'jen@example.com')]

  Subject: 2nd MOOC

  Hi!

  I hope your 2nd MOOC is going well!

  Bye :-)

- (•) To: Paul Gries <paul@example.com>, Jen Campbell <jen@example.com>,

  Subject: 2nd MOOC

  Hi!

  I hope your 2nd MOOC is going well!

  Bye :-)

9. Which of the following is **not** a special method of **object**? 1 point

- ( ) **__str__**
- (•) **__lower__**
- ( ) **__ne__**
- ( ) **__eq__**

10. Consider this code: 1 point

```
class Author:
    def __init__(self, name):
        """ (_______, str) -> NoneType """
        self.name = name
```

What should the blank (_______) in the type contract be replaced with?

- ( ) **NoneType**
- ( ) **str**
- (•) **Author**
- ( ) It is not possible to tell.

[x] I, **Shou-Yi (Ray) Hung**, understand that submitting another's work as my own can result in zero credit for this assignment. Repeated violations of the Coursera Honor Code may result in removal from this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Save Submit