Congratulations! You passed!
TO PASS 80% or higher

Keep Learning

GRADE
100%

Practice Python Assessment I

TOTAL POINTS 13

1. Which of the following list comprehensions will extract all tuples in `words` that have origins that are Greek? Select as many as apply. **2 / 2 points**

```
words= [("time", ("Noun"), ("Middle English")),
        ("take", ("Verb"), ("Gothic")),
        ("people", ("Noun"), ("Latin")),
        ("think", ("Verb", "Adjective"), ("Gothic")),
        ("work", ("Noun", "Verb", "Adjective"), ("Greek", "Gothic")),
        ("company", ("Noun"), ("Late Latin")),
        ("problem", ("Noun", "Adjective"), ("Greek")),
        ("feel", ("Verb", "Noun"), ("Old Norse")),
        ("public", ("Adjective", "Noun"), ("Latin")),
        ("aberration", ("Noun"), ("Latin")),
        ("annul", ("Verb"), ("Latin")),
        ("cacophony", ("Noun"), ("Greek")),
        ("grovel", ("Verb"), ("Old Norse")),
        ("muse", ("Noun", "Verb"), ("Greek", "Middle French")),
        ("sublime", ("Adjective", "Noun", "Verb"), ("Latin")),
        ("vilify", ("Verb"), ("Late Latin")),
        ("diffuse", ("Verb", "Adjective"), ("Latin")),
        ("elucidate", ("Verb"), ("Late Latin")),
        ("flout", ("Verb", "Noun"), ("Middle English")),
        ("grouse", ("Verb", "Noun"), ("?")),
        ("limpid", ("Adjective"), ("Latin")),
        ("nexus", ("Noun"), ("Latin")),
        ("quibble", ("Verb", "Noun"), ("?")),
        ("verbose", ("Adjective"), ("Latin")),
        ("catalyst", ("Noun"), ("?")),
        ("nocturnal", ("Adjective"), ("Late Latin")),
        ("diurnal", ("Adjective", "Noun"), ("Latin")),
        ("malapropism", ("Noun"), ("English"))]
```

- [x] greek_origins = [data for data in words if “Greek” in data[2]]

  Correct
  topics to review: list comprehensions, extracting data from tuples, conditionals

- [x] greek_origins = [data for data in words if “Greek” in data[-1]]

  Correct
  topics to review: list comprehensions, extracting data from tuples, conditionals

- [ ] greek_origins = [data for data in words if data[2] in “Greek”]
- [ ] greek_origins = [data for data in words if data[-1] == “Greek”]

2. Which of the following lines of code will sort the list of tuples called `circus` by the values stored in the fifth item in each tuple from highest to lowest? (Note that when we say fifth we mean what a human would consider fifth.) **1 / 1 point**

- [x] sorted(circus, key = lambda d: d[4], reverse = True)

  Correct
  Topics to review: sorting, key parameters, indexing with tuples

- [ ] sorted(circus, lambda z: z[4], True)
- [ ] sorted(circus, key = lambda d: d[4], reverse = False)
- [ ] sorted(circus, key = lambda c: circus[4], reverse = True)

3. Which of the following functions would create a dictionary that keeps track of the frequency of each letter in the string that is provided as input, and returns that dictionary? **2 / 2 points**

- [ ]

```
def feq_of_chars(s):
    d = {}
        for wrd in s.split():
            for char in wrd:
                if char in wrd:
                    d[char] += 1
                else:
                    d[char] = 1
            return d
```

```
def freq_of_chars(s):
    d = {}
    for wrd in s.split():
        for char in wrd:
            if char in d:
                d[char] += 1
            else:
                d[char] = 0
```

```
def freq_of_chars(s):
    d = {}
    for wrd in s.split():
        for char in wrd:
            if char not in d:
                d[char] = 1
            else:
                d[char] += 1
    return d
```

Correct

Topics to review: function definition, return statements, conditionals, dictionaries, nested for loops

```
def freq_of_chars(s):
    d = {}
    for wrd in s:
        if char not in d:
            d[char] = 0
        d[char] += 1
    return d

```

4. Which of the following while loops has an infinite loop? Select as many as apply. **1 / 1 point**

```
z = "newspaper stand? It's across the way."
t = 0
while t <= len(z):
    print(z[t])
    if z[t] in [".", ",", "!", "?"]:
        print(t)
        break
    t += 1

```

```
while False:
    input = ("enter a number")
    print(input)

```

```
x = 13.0
while True:
    print(x)
    if x > 15.0:
        break
    x += 0.5

```

```
while True:
    input = ("enter a number")
    print(input)

```

Correct

Topics to review: while loops, input, infinite loops

5. Which of the following child classes (called Magic) will properly inherit from the Character parent class while **adding** in the ability to keep a list of magic spells, where each spell is an object of the Magic class? Please note: the solution must not override the parent class attribute values and it must allow the program to set all parent class attribute values in the constructor when creating a new instance of the Magic class. **1 / 1 point**

Select as many as apply.

```
class Character():
    def __init__(self, name, height, alignment, level = 1, health = 50):
        self.name = name
        self.height = height
        self.alignment = alignment
```

```
        self.level = level
        self.health = health

    def init_skills(self, strength, dext, intel, wisdom, charm):
        self.strength = strength,
        self.dexterity = dext
        self.intelligence = intel
        self.wisdom = wisdom
        self.charm = charm

    def backstory(self, history):
        try:
            if self.backstory[-1] != " ":
                self.backstory = self.backstory + " " + history
            else:
                self.backstory += history
        except:
            self.backstory = history

```

- [ ]
```
class Magic(Character):

    def __init__(self, spells):
        self.spells = spells

    def add_Spell(self, spell):
        self.spells.append(spell)

```

- [x]
```
class Magic(Character):

    def __init__(self, name, height, alignment, level = 1, health = 50, spells = []):
        Character.__init__(self, name, height, alignment, level, health)
        self.spells = spells

    def add_Spell(self, spell):
        self.spells.append(spell)

```

✓ **Correct**

Topics to review: Classes, Inheritance, class methods, lists, try/except

- [ ]
```
class Magic(Character):

    def __init__(self, name, height, alignment, level = 1, health = 50, spells = []):
        Character.__init__(self, name, height, alignment, level = 1, health = 50)
        self.spells = spells

    def add_Spell(self, spell):
        self.spells.append(spell)

```

- [ ]
```
class Magic():

    def add_Spell(self, spell):
        try:
            self.spells.append(spell)
        except:
            self.spells = [spell]

```

- [x]
```
class Magic(Character):

    def add_Spell(self, spell):
        try:
            self.spells.append(spell)
        except:
            self.spells = [spell]
```

✓ **Correct**

Topics to review: Classes, Inheritance, class methods, lists, try/except

6. Which of the following functions is **least likely** to break if there is an error when the program makes a request to "exampleapi.com"? Note that we expect for py_data to hold a dictionary.

1 / 1 point

- [ ]
```
def convert_data(title):
    d = {"t": title}
    baseurl = "exampleapi.com"
    data = requests.get(baseurl, params = d)
    py_data = json.loads(data.txt)
    if type(py_data) == type({}):
        return py_data
    else:
        return py_data

```

```python
def convert_data(title):
    d = {"t": title}
    baseurl = "exampleapi.com"
    data = requests.get(baseurl, params = d)
    py_data = json.loads(data.txt)
    return py_data
```

```python
def convert_data(title):
    d = {"t": title}
    baseurl = "exampleapi.com"
    try:
        data = requests.get(baseurl, params = d)
        py_data = json.loads(data.txt)
        return py_data
    except:
        return None
```

```python
def convert_data(title):
    d = {"t": title}
    baseurl = "exampleapi.com"
    data = requests.get(baseurl, params = d)
    try:
        py_data = json.loads(data.txt)
        return py_data
    except:
        return None
```

Correct

Topics to review: try/except, requesting data through APIs, loading json data

This answer is least likely to break because the request and conversion of data are stored in the try/except. If something went wrong with the data collection or the conversion to python, then the program would move to the exception.

7. Please fill in the blank so that the following code successfully iterates through the data stored in "makeup_products" and extracts the total number of shades for items that are meant just for "Lips", storing the total in a variable called "total_count_lip_shades".

2 / 2 points

```python
makeup_products = {"Products": [
                    {"Primer": {
                                "Shades": 15,
                                "Styles": 5,
                                "Location": "Face"
                               }
                    },
                    {"Lipstick": {
                                "Shades": 48,
                                "Styles": 3,
                                 "Location": "Lips"
                               }
                    },
                    {"Lip liner": {
                                "Shades": 32,
                                "Styles": 4,
                                 "Location": "Lips"
                                }
                    },
                    {"Blush": {
                                "Shades": 13,
                                "Styles": 2,
                                "Location": "Face"
                               }
                    },
                    {"Eye Liner": {
                                "Shades": 14,
                                "Styles": 7,
                                "Location": "Eye"
                                }
                    },
                    {"Travel Makeup Kit": {
                                "Shades": "N/A",
                                "Styles": 3,
                                "Location": "Face, Lips, Eye"
                                }
                    },
                    {"Chapstick": {
                                "Shades": 3,
                                "Styles": 7,
```

```
if item[product]["Location"] == "Lips":
```

Correct

Topics to review: conditionals, nested data extraction, dictionaries.

For this particular question, we needed to determine what key to use for each dictionary stored in item. Once we know what key to use, we need to dig deeper to see what key value pair mentions where the product should be worn. By looking at the dictionary, we can see that that information is stored in the key "Location".

8. Assume that a json-structured string has been stored in a file named “improv_data.json”. Fill in the blank to convert the json string into a python object and store it in the variable improv_py. **1 / 1 point**

```
import json
improv_f = open("improv_data.json", "r").read()
improv_py = # your code would finish this statement
improv_f.close()
```

json.loads(improv_f)

Correct

For this problem, it would take the string assigned to improv_f and use json.loads to load the string into a python object.

Topics to review: json, reading from files

9. [Practice Jupyter Notebook] How many lines are in the file list_of_books.txt? **1 / 1 point**

5

Correct

This is correct. To determine this, you could count the number of \n that appear in the file, check the length of the list produced by readlines, or count the number of times that the for loop iterates through the file.

10. [Practice Jupyter Notebook] How many copies of Cathy O'Neil's Weapons of Math Destruction are available after hatcher has been defined? **1 / 1 point**

1

Correct

This is correct. We can print out the the books attribute and see in the output that the value for available is 1. We could also dig specifically into the nested data and print out the value assigned to the key available, and see that that is 1.