

CS303E Week 11B Worksheet: Tuples, Sets and Dictionaries

Name: _____ EID: _____

Read the questions carefully, and answer each question in the space provided. Use scratch paper to do your work and then copy your answers neatly and legibly onto the test paper. Only answers recorded on the test paper will be graded.

1. (11 points: 1 point each) The following are true/false questions. **Write either T or F in the boxes at the bottom of page 1.** If there's any counterexample, it's false.
 - (a) If a list in Python contains multiple identical items and you try to convert it to a set, the program will crash.
 - (b) If you try to access a key that does not exist in a dictionary, Python will return `None`.
 - (c) Using curly braces `{}` creates an empty set.
 - (d) The dictionary comprehension `{str(x):0 for x in range(1, 6)}` is equivalent to the dictionary `{"1":0, "2":0, "3":0, "4":0, "5":0}`.
 - (e) The `"in"` operator can be used to check for the presence of an element in a tuple, but it doesn't work on sets because sets have no order.
 - (f) The symmetric difference of two sets in Python can also be expressed as the union of the two sets, minus their intersection.
 - (g) The `.pop()` method for dictionaries in Python removes the last added key-value pair of the dictionary and returns it.
 - (h) When iterating over a dictionary using a `for` loop, the loop variable refers to the keys of the dictionary.
 - (i) You can loop over sets, but you cannot loop over tuples.
 - (j) Assuming `s1` and `s2` are both sets, the expression `s1.difference(s2)` can return a different result from `s2.difference(s1)`.
 - (k) Both lists and tuples in Python support adding elements after creation, but unlike lists, elements in a tuple cannot be changed once added.

a	b	c	d	e	f	g	h	i	j	k
										.

Questions 2-12 are multiple choice. Each counts 2 points. **Write the letter of the BEST answer in the box on the next page. Please write your answer in UPPERCASE. Each problem has a single answer.**

2. Which of the following is a valid way to create a tuple containing the integer 1?
 - A. `harry = 1,`
 - B. `liam = (1,)`
 - C. `niall = (1)`
 - D. `louis = {1,}`
 - E. All of the above.
 - F. A and B

3. Which approach will correctly delete key/value pairs from a dictionary (`myDict`) where the value is less than some threshold `x`?
 - A.

```
for value in myDict:
    if not value >= x:
        del value
```
 - B.

```
myKeys = list(myDict.keys())
for key in myKeys:
    if myDict[key] < x:
        del myDict[key]
```
 - C.

```
for i in range(len(myDict)):
    if myDict[i] < x:
        myDict.pop(i)
```
 - D.

```
removeKeys = [val for key, val in myDict.items() if val < x]
for key in removeKeys:
    del myDict[key]
```

4. If `s1` and `s2` are both sets, and `s1.issuperset(s2)` is `False`, which of the following expressions will always evaluate to `True`?
 - A. `s1.difference(s2) == s1`
 - B. `s1.symmetric_difference(s2) != set()`
 - C. `s1.intersection(s2) == set()`
 - D. `len(s2) >= len(s1)`

5. What will happen if you try to convert a dictionary into a list using `list()`?
 - A. An error is raised.
 - B. It will create a list of the dictionary's keys.
 - C. It will create a list of the dictionary's values.
 - D. It will create a list of the dictionary's key-value pairs as tuples.

6. What can the `tuple()` constructor be used for?
- A. Creating an empty tuple.
 - B. Converting a single number into a one-item tuple.
 - C. Converting other types such as strings or lists to tuples.
 - D. All of the above.
 - E. A and C.
7. Assuming `set1` and `set2` are both sets, how can you obtain a set containing only the elements present in one (not both)?
- A. `set1.difference(set2).union(set2.difference(set1))`
 - B. `set1.symmetric_difference(set2)`
 - C. `set1.union(set2).difference(set1.intersection(set2))`
 - D. `set1.difference(set2).union(set2 - set1)`
 - E. All of the above.
 - F. A and C.
8. What is possible output of the following code?

```
pleakley = {"lilo", "stitch"}
jumba = pleakley
jumba.add("nani")
bubbles = set(pleakley)
bubbles.remove("stitch")
print(pleakley, jumba, bubbles)
```

- A. `{"nani", "stitch", "lilo"} {"nani", "stitch", "lilo"} {"nani", "lilo"}`
 - B. `{"stitch", "lilo"} {"nani", "stitch", "lilo"} {"lilo"}`
 - C. `{"stitch", "lilo"} {"nani", "stitch", "lilo"} {"nani", "lilo"}`
 - D. `{"nani", "stitch", "lilo"} {"nani", "stitch", "lilo"} {"lilo"}`
 - E. None of the above.
9. Which of the following correctly demonstrates tuple unpacking in Python?
- A. `dreamland, eatsAlot = ("kirby", "dedede", "metaKnight")`
 - B. `(arcanine, chansey) = "jigglypuff", "jynx", "clefairy"`
 - C. `blathers, ableSisters = ("isabelle", "tomNook"), ("rover", "kkSlider")`
 - D. `courage, (wisdom, power) = ("link", ("zelda", "ganon"))`
 - E. C and D.
 - F. B, C and D.

10. Which of the following correctly represents the comparison rules for tuples, sets, and dictionaries in Python?
- Tuples are compared element-wise using `<` and `>`. Sets can be tested to see if one is a subset or superset of another. Dictionaries can only be compared for equality.
 - Tuples are compared based on their length. Sets support element-wise comparisons using `>` and `<`. Dictionaries cannot be compared.
 - Tuples can only be compared for equality. Sets are compared based on their length. Dictionaries can only be compared for equality.
 - Tuples are compared element-wise using `<` and `>`. Sets can be tested to see if one is a subset or superset of another. Dictionaries cannot be compared.
11. What happens when you try to add two dictionaries using the `+` operator?
- The dictionaries are merged, and if there are overlapping keys, the values from the second dictionary overwrite those from the first.
 - The dictionaries are merged, and if there are overlapping keys, their values are summed.
 - It creates a list containing both dictionaries as elements.
 - It raises an error.
12. Which of the following statements about tuples, sets, and dictionaries in Python is false?
- Slicing a tuple will not raise an error.
 - Dictionary keys can be associated with multiple values.
 - The keys in a dictionary must be of immutable types, such as strings or tuples.
 - For any set `s1`, `s1.issuperset(set())` will evaluate to `True`.
 - The `"in"` operator applied to dictionaries searches keys and not values.

2	3	4	5	6	7	8	9	10	11	12

The following 10 questions require you to trace the behavior of some Python code and identify the output of that code. For each question, write the output for the code segment on the provided line.

13. (3 points)

```
def avatar(krew):
    krew += ("tenzinAir", "bolinEarth", "makoFire")
    return krew

otp = ("korra", "asami")
avatar(otp)
for i in range(len(otp)):
    if i % 2:
        print(otp[i][:5], end = "")
    else:
        print(otp[i][:4], end = "")
```

14. (3 points)

```
gaang = ("aang", "katara", "sokka")
gaang += ("toph", "zuko")
print(gaang, end = " ")
gaang -= ("zuko", "aang")
print(gaang, end = " ")
```

15. (3 points)

```
marioKart = {"donkeyKong": "goodie", "wario": "baddie", \
            "luigi": "goodie", "bowser": "baddie"}
marioParty = {char: align for align, char in marioKart.items()}
print(marioParty)
```

16. (3 points)

```
teenTitans = {"robin", "starfire", "raven", "beastboy", "cyborg"}
girlsRule = {"starfire", "raven"}
boysDrool = {"robin", "beastboy", "cyborg"}

azarath = teenTitans.difference(boysDrool)
silkie = teenTitans.symmetric_difference(boysDrool)

if azarath == silkie:
    print(teenTitans.intersection(girlsRule))
else:
    print(boysDrool.union(girlsRule))
```

17. (3 points)

```
monsterHigh = {"frankie", "draculaura", "clawdeen", "cleo"}
ghoulSpirit = {"frankie", "lagoon", "ghoulia", "abbey", "spectra", "clawdeen"}
ghoulPower = monsterHigh.symmetric_difference(ghoulSpirit)
sixInchHeels = monsterHigh + ghoulSpirit.difference(monsterHigh)
print(ghoulPower == sixInchHeels)
```

18. (3 points)

```
blastFromPast = ("nancyDrew", ["peppaPig", "goosebumps"], "junieBjones")
blastFromPast[1][0] = "magicSchoolBus"
print(blastFromPast)
```

19. (3 points)

```
simba = (3, 7)
timon, pumbaa = simba
timon += 1; pumbaa += 1
simba += (2, 5)
print(timon, pumbaa, simba)
```

20. (3 points)

```
spotifyPlaylist = {"partyInUSA": "miley", "hitMeBaby": "britney",
                  "want4Christmas": "mariah", "theClimb": "miley",
                  "obsessed": "mariah", "circus": "britney",
                  "singleLadies": "beyonce"}
print({spotifyPlaylist.pop(song) for song in list(spotifyPlaylist.keys())})
```

21. (3 points)

```
overwatch = {"soldier76": {"hp":200, "ult":300, "dmg":100},
             "mercy":      {"hp":200, "ult":250, "dmg":60},
             "dva":        {"hp":600, "ult":400, "dmg":80} }
patch =      {"soldier76": {"hp":50, "ult":-20, "dmg":10},
             "mercy":      {"hp":25, "ult":10, "dmg":10},
             "dva":        {"hp":0, "ult":20, "dmg":5} }

for hero, changes in patch.items():
    for stat, adjust in changes.items():
        overwatch[hero][stat] = overwatch[hero][stat] + adjust
# i didnt realize how much this is to write...
# i'll probably tweak this question, just treat
# it as a fun exercise for now =)
print(overwatch)
```

22. (3 points)

```
mister = [3, 5, 2, 0, 3, 8, 1, 6, 9, 4, 2, 7, 3, 5, 9, 7]
taterTot = set()
for pumpkin in mister:
    if pumpkin % 3 == 0:
        taterTot.add(pumpkin)
    elif pumpkin % 2 == 0:
        taterTot.add(pumpkin // 2)
    else:
        taterTot = taterTot.difference({pumpkin // 2})
print(taterTot)
```