Computer Science Honors — Test 2 — Name: ____________________

Fall 2015 — Mr. Alwin Tareen

Part I. (53 points) Solve each of the following problems. For the multiple choice problems, select the correct answer by placing an "X" in the box beside it.

Quick Reference

- `prices = {}` creates an empty dictionary
- `prices = {'apple':1, 'pear':2}` creates a non-empty dictionary
- `prices['apple']` returns the value that's mapped to by `'apple'`
- `prices['apple'] = 5` maps the value `5` to `'apple'`. Overwrites the previous value.
- `del prices['apple']` deletes the mapping with the key `'apple'` from `prices`
- `len(prices)` returns the number of entries in `prices`
- `x in prices` checks whether the key `x` is in the dictionary `prices`
- `prices.keys()` returns a list of all the keys in the dictionary.
- `prices.values()` returns a list of all the values in the dictionary.

(1pt) **1.** How would you create an empty dictionary named `cost`? [ ] 1 pt

☐ `cost = {}` ☐ `cost = []` ☐ `cost = ()` ☐ `cost = <>`

(1pt) **2.** How would you find the number of entries in the dictionary `total`? [ ] 1 pt

☐ `num(total)` ☐ `len(total)` ☐ `amount(total)` ☐ `entries(total)`

(1pt) **3.** How would you return a list of all the keys in the dictionary `goods`? [ ] 1 pt

☐ `allKeys.goods` ☐ `keys-->goods` ☐ `goods.keys()` ☐ `keys[goods]`

(1pt) **4.** How would you return a list of all the values in the dictionary `stock`? [ ] 1 pt

☐ `values stock` ☐ `stock&values` ☐ `values{stock}` ☐ `stock.values()`

(1pt) **5.** Which method in a dictionary gives you a list of the values in the dictionary? [ ] 1 pt

☐ `each()` ☐ `items()` ☐ `values()` ☐ `keys()`

(2pts) **6.** What is the purpose of the second parameter of the `get()` method for Python dictionaries? [ ] 2 pts

☐ To provide a default value if the key is not found

☐ An alternate key to use if the first key cannot be found

☐ The value to retrieve

☐ The key to retrieve

[ ] 7 pts

(2pts) **7.** How are Python dictionaries different from Python lists? [2 pts]

- ☐ Python lists store multiple values, and dictionaries store a single value
- ☐ Python lists can store strings, and dictionaries can only store words
- ☐ Python lists are indexed using integers, whereas dictionaries can use strings as indexes
- ☐ Python dictionaries are a collection, and lists are not a collection

(2pts) **8.** What would the following Python code print out? [2 pts]

```
stuff = dict()
print stuff['candy']
```

- ☐ The program would fail with a traceback
- ☐ `-1`
- ☐ `candy`
- ☐ `0`

(2pts) **9.** What would the following Python code print out? [2 pts]

```
stuff = dict()
print stuff.get('candy', -1)
```

- ☐ The program would fail with a traceback
- ☐ `-1`
- ☐ `candy`
- ☐ `0`

(2pts) **10.** What is a common use of Python dictionaries in a program? [2 pts]

- ☐ Computing an average of a set of numbers
- ☐ Splitting a line of input into words using a space as a delimiter
- ☐ Building a histogram counting the occurrences of various strings in a file
- ☐ Sorting a list of names into alphabetical order

(2pts) **11.** In the following Python code, what does the `for` loop iterate through? [2 pts]

```
x = dict()
for y in x:
```

- ☐ It loops through the values in the dictionary
- ☐ It loops through the keys in the dictionary
- ☐ It loops through all the dictionaries in the program
- ☐ It loops through the integers in the range from zero through the length of the dictionary

(2pts) **12.** What are the keys in the following Python dictionary? `d = {'john':40, 'peter':45}` [2 pts]

- ☐ `'john' and 'peter'`
- ☐ `'john', 40, 45, and 'peter'`
- ☐ `40 and 45`
- ☐ `40 and 'peter'`

[12 pts]

(2pts) **13.** What will be the output of the following Python code?

```
d = {‘john’:40, ‘peter’:45}
print ‘john’ in d
```

☐ `True`
☐ `None`
☐ `False`
☐ This program fails with a traceback

2 pts

(2pts) **14.** What will be the output of the following Python code?

```
d = {‘john’:40, ‘peter’:45}
print d[‘john’]
```

☐ `45`
☐ `‘peter’`
☐ `40`
☐ `‘john’`

2 pts

(2pts) **15.** Given the dictionary: `prices = {‘banana’:4, ‘apple’:2, ‘orange’:1.5, ‘pear’:3}`
How would you look up the price of an `apple`?

☐ `prices[‘apple’]`
☐ `prices.retrieve(apple)`
☐ `keyCorrespond{apple}`
☐ `getValue(‘apple’)`

2 pts

(2pts) **16.** Given the dictionary: `stock = {‘banana’:6, ‘apple’:0, ‘orange’:32, ‘pear’:15}`
How would you subtract 1 from the stock of `orange`?

☐ `[orange].reduce(1)`
☐ `stock.orange.minus.1`
☐ `stock[‘orange’] -= 1`
☐ `orange subtraction 1`

2 pts

(2pts) **17.** Given the dictionary: `cheese = {‘swiss’:3, ‘cheddar’:7, ‘gouda’:4}`
Which of the following statements checks whether or not `‘swiss’` is in the dictionary `cheese`?

☐ `cheese.containsValue(swiss)`
☐ `cheese -> swiss`
☐ `cheese valueExcluding(cheddar, gouda)`
☐ `‘swiss’ in cheese`

2 pts

(2pts) **18.** What is the output of the following code:

```
food = {‘pizza’:3}
food[‘fries’] = 10
print food
```

Answer:

2 pts

12 pts

(2pts) **19.** What is the output of the following code: | 2 pts

```
treasure = {‘gold’:50, ‘silver’:100}
‘gold’ in treasure
```

Answer:

(2pts) **20.** What is the output of the following code: | 2 pts

```
breakfast = {‘coffee’:2, ‘eggs’:4, ‘bacon’:7}
if breakfast[‘eggs’] > 3:
    print ‘Yum!’
else
    print ‘Still hungry!’
```

Answer:

(2pts) **21.** What is the output of the following code: | 2 pts

```
inventory = {
    ‘pocket’:‘lint’,
    ‘canteen’:‘water’,
    ‘pouch’:‘flint’,
    ‘backpack’:[‘shovel’, ‘bedroll’, ‘rope’]
}
print inventory[‘backpack’]
```

Answer:

(3pts) **22.** What is the output of the following code: | 3 pts

```
inventory = {
    ‘gold’:500,
    ‘pouch’:‘flint’,
    ‘backpack’:[‘shovel’, ‘bedroll’, ‘rope’]
}
print ‘silver’ in inventory
print len(inventory)
print inventory[‘pouch’]
```

Answer:

Answer:

Answer:

(2pts) **23.** What is the output of the following code: | 2 pts

```
fortune = {‘gold’:500}
fortune[‘gold’] += 50
print fortune
```

Answer:

(2pts) **24.** What is the output of the following code: | 2 pts

```
inventory = {
    ‘gold’ : 500,
    ‘backpack’ : [‘xylophone’, ‘dagger’, ‘bedroll’]
}
inventory[‘backpack’].sort()
print inventory[‘backpack’]
```

Answer:

13 pts

(2$^{\text{pts}}$) **25.** What is the output of the following code:

```
grocery = {‘kiwi’:5, ‘grape’:12}
del grocery[‘kiwi’]
print grocery
```

Answer:

2 pts

(3$^{\text{pts}}$) **26.** Consider the following two dictionaries that model a simple grocery store. `prices` gives the cost of each item, and `stock` indicates the quantity of each item in the store.

```
prices = {‘banana’:4, ‘apple’:2, ‘orange’:1.5, ‘pear’:3}
stock = {‘banana’:6, ‘apple’:0, ‘orange’:32, ‘pear’:15}
```

Write a program that calculates the total value of all the items in the store. *Hint:* Find a way to multiply prices and stock.

3 pts

5 pts

(4pts) **27.** Consider the following dictionary that models a student's grade report.

```
albert = {
        'name' : 'albert',
        'homework' : [99, 98, 100],
        'quizzes' : [89, 95],
        'tests' : [91, 93]
        }
```

4 pts

Also, consider the following function that computes the average of a list:

```
def average(numbers):
    return sum(numbers) / len(numbers)
```

Write a program that computes a students' final grade using a *weighted average*. The weights should be defined as follows:

- Homework is worth 20%.
- Quizzes are worth 30%.
- Tests are worth 50%.

4 pts