MQF Exam 1 Spring 2017 Name_____

Return your question sheets and blue book(s)

Concept questions (1 point each)

1. To create a constructor of a class call Truck in Python which method of the following shall be defined?

   a. `Truck()` b. `__init__` c. `Create()` d. `__construct__`

2. Which function provides a way to check an object's type?

   a. `istype()` b. `isinstance()` c. `showtype()` d. `instance()`

3. The __str__() method of the object class returns the _____ if not overrided.

   a. name of the class and its attributes
   b. name of the class and its id
   c. name of the object and its methods
   d. arguments listed in the call

4. The isdigit() method of a string returns

   a. the digits that are in the string
   b. the string if it only contains digits
   c. true if the string contains only digits
   d. true if the string contains only digits and a decimal point

5. The join() method of a list can be used to combine

   a. the items in the list into a string
   b. the items in the list into a string that's separated by delimiters
   c. two or more lists
   d. two or more strings into a list

6. In a method of a class, the first parameter, which is usually named self, refers to the current

   a. object
   b. class
   c. method
   d. parameter

7. Given a class named Customer, which of the following creates a Customer object and assigns it to the variable named cust1:

   a. `cust1 = new Customer()`
   b. `cust1 = Customer()`
   c. `cust1 = Customer.init()`
   d. `cust1 = Customer.create()`

8. A dictionary stores a collection of

   a. ordered items
   b. unordered items
   c. mutable items
   d. immutable items

9. Each item in a dictionary is

   a. a key/value pair
   b. a sequence
   c. a string
   d. a list

10. To avoid a KeyError when using the pop() method of a dictionary, you can

a. use the optional second argument to supply the correct key
b. use the optional second argument to supply a default value
c. use the exists keyword to check whether the key exists before you call the pop() method
d. use the del keyword to check whether the pop() method can delete the key without a KeyError

11. Which of the following code snippets will result in this display:

```
Countdown...
5...
4...
3...
2...
1...
Blastoff!
```

a.
```
counting = "5...4...3...2...1"
print("Countdown...")
for char in counting:
    print(char + "...")
print("Blastoff!")
```

b.
```
counting = "54321"
for char in counting:
    print("Countdown...")
    print(char + "...")
    print("Blastoff!")
```

c.
```
counting = "54321"
print("Countdown...")
for char in counting:
    print(char + "...")
print("Blastoff!")
```

d.
```
counting = 54321
print("Countdown...")
for char in counting:
    print(char + "...")
print("Blastoff!")
```

12. How many items does the following dictionary contain?

```
flowers = {"red": "rose", "white": "lily", "yellow": "buttercup", "purple":
"tulip" }
```

a. 1 b. 3 c. 4 d. 8

Consider the following program segment

```
1.   flowers = {"red": "rose", "white":"lily","yellow":"buttercup","purple":
"tulip" }
2.   print(flowers)
3.   flowers["blue"] = "carnation"
4.   print(flowers)
5.   print("This is a red flower:", flowers.get("red", "none"))
6.   key = "white"
7.   if key in flowers:
8.       flower = flowers[key]
9.       print("This is a", key, "flower:", flower)
10.  key = "green"
11.  if key in flowers:
12.      flower = flowers[key]
13.      del flowers[key]
14.      print(flower + " was deleted")
15.  else:
16.      print("There is no " + key + " flower")
```

13. Refer to the code above: Which of the following represents a key/value pair for the dictionary named flowers defined on line 1?

a. `lily/white`
b. `red/rose`
c. `blue/carnation`
d. `yellow/flower`

14. Refer to code above: What would the print statement on line 9 display?

a. `This is a white flower: white`

b. `This is a white flower: lily`

c. `This is a lily flower: white`

d. `This is a lily flower: lily`

15. What is the value of s3 after the code that follows is executed?

```
s1 = "abc def ghi";
s2 = s1[1:5]
s3 = s2.replace('b', 'z')
print(s3)
```

a. `bc d`
b. `zc d`
c. `abc d`
d. `azc d`
e. `zc de`

16 Which of the following will display this result?

```
B = 66
```

a. `print("B = ", ord("B"))`
b. `print("B = ", char("B"))`
c. `print("B = ", ascii("B"))`
d. `print(ord(66))`

17. To determine the length of a string that's in a variable named city, you can use this code:

a. `len(city)`
b. `city.len()`
c. `length(city)`
d. `city.length()`

18. You can use the split() method to split a string into a list of strings based on a specified

a. word
b. character
c. delimiter
d. punctuation mark

19. To retrieve the fifth character in a string that's stored in a variable named city, you can use this code:

a. `city(5)`
b. `city[3]`
c. `city(4)`
d. `city[4]`

20. To access the substring from character 2 to character 4 of a string that's stored in a variable named message, you can use this code:

a. `first_three = message[0:3]`
b. `first_three = message[2:5]`
c. `first_three = message.slice(2:4)`
d. `first_three = message.split(0:3)`

21. Given the following code, what will be displayed after the code executes?

```
name = "Mervin the Magician"
words = name.split()
print(words[0] + ", you are quite a " + words[2].lower())
```

a. `Mervin`
`, you are quite a`
`magician`
b. `Mervin, you are quite a`
`Magician`
c. `Mervin, you are quite a`
`magician`
d. `Mervin`
`, you are quite`
`the magician`

22. If `word = "a horse"`, which of the following snippets of Python code will display this result?

```
a horsea horse!  My kingdom for a horse!
```

a. `print((word * 2) + "! My kingdom for " + word + "!")`
b. `print((word + "! " + " My kingdom for " + word + "!") * 2)`
c. `print(word * 2 + " My kingdom for " + word + "!")`
d. `print((word + "! ") * 2 + " My kingdom for " + word + "!")`

23. Which of the Python examples that follow can *not* be used to create a string named n2?

a. `n2 = "817542235"`
b. `numbers = "817542235"`
`n2 = numbers.replace("2", "")`
c. `numbers = [8, 17, 54, 22, 35]`
`n2 = "".join(numbers)`
d. `numbers = ["8", "17", "54", "22", "35"]`
`n2 = "".join(numbers)`

24. Which of the following is not a proper way to import module from Python? For example we want to use sin() of math module

a. `import math as m`
b. `from math import *`
c. `math import`
d. `Import math`

25. The finally clause of a try statement

a. is required
b. is executed whether or not an exception has been thrown
c. can be used to display more information about an exception
d. can be used to recover from an exception

26. To throw an exception with Python code, you use the

a. raise statement
b. throw statement
c. built-in throw() function
d. build-in raise() function

27. The _____ method adds an item to the end of a list.

a. pop() b. append() c. insert() d. index()

28. To insert the item "melon" after "grapes" in the following list, you would use which of these methods?

```
fruit = ["apple", "banana", "grapes", "mangos", "oranges"]
```

a. `fruit.pop("melon", 3)`
b. `fruit.insert("melon", 3)`
c. `fruit.insert(3, "melon")`
d. `fruit.append(3, "melon")`

29. What will this loop display?

```
sum = 0
for i in range(0, 15, 5):
    sum += i
print(sum)
```

a. `30`
b. `50`
c. `0, 5, 10, 15`
d. `None of the above`

30. What line number of the following code contains an error and what type of error is it?

```
1.  def sales_tax(amt):
2.      sale = amt + (amt * .06)
3.      return amt
4.
5.  def main():
6.  print("Welcome to the 6% tax calculator!\n")
7.      total = int(input("Please enter the total amount: "))
8.      print("The total amount after tax is: ", sales_tax(total))
```

a. line 6, syntax error
b. line 1, syntax error
c. line 3, runtime error
d. line 8, logic error

Short answers (5 points each)

1. What is the output of the following Python script

```
a = ['cbc', 'is', 'easy']
b = []
for c in a:
    for d in c:
        b.append(d)
print(b)
```

2. (list comprehension)

   What is the output of the following print statement?

```
print ([(c, c*c) for c in range(5)])
```

3. (String method join)

   What is the output of the following Python script?

```
>>> "".join(['cat', 'dog', 'rabbit'])
```

4. (set operations)
   What are the outputs of the following?

```
a={1,3,5,7,9}
b={1,2,3,4,5,6,7}
print(a | b)
print(a & b)
print(a - b)
print(b - a)
print(a ^ b)
```

Programming (10 points each)

1. Write a function and call it contains. The function head is as the following

```
def contains(data, target):
```

Write the function body so that when the following statements are executed

```
lst = [1,3,5,"great", "small", 321.567]
print (contains(lst, 5))
print (contains(lst, "big"))
```

The corresponding outputs are as the following

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
True
False
>>> |
Ln: 15 Col: 4
```

2. Write a recursive Fibonacci function and a main program which will use this function and output Fibonacci numbers from 20 to 30. I.e., compute Fib(20), Fib(21),… Fib(30). The output shall like the following. Note the Fibonacci (20) is 6765 and Fibonacci (30) is 832040.

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> ============================ RESTART ============================
>>>
6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040, ...
>>>
Ln: 176 Col: 4
```

3. Define class Vector which overload some operators.

```
class Vector:
  def __init__(self, d):
    if isinstance(d, int):
      self._coords = [0] * d
    else:
      try:                                    # we test if param is iterable
        self._coords = [val for val in d]
      except TypeError:
        raise TypeError('invalid parameter type')
  def __len__(self):
    return len(self._coords)
  def __getitem__(self, j):
    return self._coords[j]  #return jth coordinate of vector.
  def __setitem__(self, j, val):
    self._coords[j] = val   #set jth coordinate of vector.
  def __add__(self, other):
    """Return sum of two vectors."""
  def __eq__(self, other):
    """Return True if vector has same coordinates as other."""
  def __ne__(self, other):
    """Return True if vector differs from other."""
  def __str__(self):
    return '<' + str(self._coords)[1:-1] + '>'  # adapt list representation
  def __neg__(self):
    """Return copy of vector with all coordinates negated."""
  def __lt__(self, other):
    """Compare vectors based on lexicographical order."""
  def __le__(self, other):
    """Compare vectors based on lexicographical order."""
```

Write code for __add__, __eq__, __ne__, __neg__, __lt__, __le__ functions.

Some test runs using the overloaded operators.

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> u = Vector (5)
>>> u
<__main__.Vector object at 0x03CBA830>
>>> print(u)
<0, 0, 0, 0, 0>
>>> v =  Vector ([1,2,3,4,5])
>>> print(v)
<1, 2, 3, 4, 5>
>>> u + v
<__main__.Vector object at 0x038BD6B0>
>>> print(u+v)
<1, 2, 3, 4, 5>
>>> u = v + v
>>> print(u)
<2, 4, 6, 8, 10>
>>> u < v
False
>>> u != v
True
>>> v != v
False
>>> v[4] = 100
>>> print(v)
<1, 2, 3, 4, 100>
Ln: 36 Col: 4
```

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>> print(v)
<1, 2, 3, 4, 100>
>>> print(u)
<2, 4, 6, 8, 10>
>>> print(u[3])
8
>>> print(v[4])
100
>>> w = -v
>>> print(w)
<-1, -2, -3, -4, -100>
>>> print (u - v)
Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    print (u - v)
TypeError: unsupported operand type(s) for -: 'Vector' and 'Vector'
>>> print(u + (-v))
<1, 2, 3, 4, -90>
>>> u < (-v)
False
>>> print(-v)
<-1, -2, -3, -4, -100>
>>>
Ln: 72 Col: 4
```

4. Write a Python program which will ask the user to enter his/her age. If the age is not integer, your program will say “Invalid input”. If enter integer but it is negative, then your program will say age must be positive. If user enters positive integer, your program will end by saying Bye…
   In a sense, your program will keep on asking user until user enter a valid age which is positive integer.

   The following is a test run screen shot

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
Enter your age in years: -23
Your age must be positive
Enter your age in years: -34.5
Invalid response
Enter your age in years: fifty
Invalid response
Enter your age in years: I am pretty and young
Invalid response
Enter your age in years: okay
Invalid response
Enter your age in years: 23
Bye ...
>>> |
Ln: 31 Col: 4
```

5. Write a class call it CreditCard. A credit card has following attributes:

   balance (for example 500)

   customer (for example, John Smith)

   bank (for example, Wells fargo)

   account (account number a string of 16 characters)

   limit (in dollar, so integer will do in this example)

In this class, there is a constructor, five(5) getters: *get_customer, get_balance, get_bank, get_account, get_limit*,

Two utility functions: *charge*, and *make_payment*

The following are screen shots for test driver and its output

```
wallet = []
wallet.append(CreditCard('John Bowman', 'California Savings',
                         '5391 0375 9387 5309', 2500) )
wallet.append(CreditCard('John Bowman', 'California Federal',
                         '3485 0399 3395 1954', 3500) )
wallet.append(CreditCard('John Bowman', 'California Finance',
                         '5391 0375 9387 5309', 5000) )

for val in range(1, 17):
  wallet[0].charge(val)
  wallet[1].charge(2*val)
  wallet[2].charge(3*val)

for c in range(3):
  print('Customer =', wallet[c].get_customer())
  print('Bank =', wallet[c].get_bank())
  print('Account =', wallet[c].get_account())
  print('Limit =', wallet[c].get_limit())
  print('Balance =', wallet[c].get_balance())
  while wallet[c].get_balance() > 100:
    wallet[c].make_payment(100)
    print('New balance =', wallet[c].get_balance())
  print()
```

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
Customer = John Bowman
Bank = California Savings
Account = 5391 0375 9387 5309
Limit = 2500
Balance = 136
New balance = 36

Customer = John Bowman
Bank = California Federal
Account = 3485 0399 3395 1954
Limit = 3500
Balance = 272
New balance = 172
New balance = 72

Customer = John Bowman
Bank = California Finance
Account = 5391 0375 9387 5309
Limit = 5000
Balance = 408
New balance = 308
New balance = 208
New balance = 108
New balance = 8
Ln: 84 Col: 4
```

Extra credit optional (each 3 points)

E1. Write a Python function prefix_sum which will compute the prefix sum of a list. A prefix sum list of a list A=[a1, a2, a3, a4, ..] can be defined as [a1, a1+a2, a1+a2+a3, a1+a2+a3+a4, …]. The following is an example of test run of this prefix_sum function

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> prefix_sum([1,2,3,4,5])
[1, 3, 6, 10, 15]
>>> prefix_sum([5,4,3,1])
[5, 9, 12, 13]
>>> prefix_sum([2,4,6,8,10])
[2, 6, 12, 20, 30]
>>> |
Ln: 15 Col: 4
```

E2. Write a Python program which will generate a random number and ask the user to guess it. Assume the number falls between 1 and 10. If user guess right, your program will say 'Good job', otherwise your program will continue to ask the user to guess again.