Quiz 01

Due Jan 29 at 10pm **Points** 10 **Questions** 6 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	501 minutes	10 out of 10

Correct answers are hidden.

Score for this quiz: **10** out of 10
Submitted Jan 28 at 11:33pm
This attempt took 501 minutes.

Question 1 — 2 / 2 pts

When is it appropriate to write a new function? (choose all that apply)

- [x] To encapsulate a feature
- [x] To split long functions into smaller functions
- [x] To make code more readable
- [x] To make code more reusable
- [ ] To increase my total LOC (Lines Of Code) to make my productivity look higher

Question 2

2 / 2 pts

I want to read a number from the user and then add 1. Explain why Python complains about the following code. Fix the code so it runs properly.

```
>>> resp = input('Pick a number: ')
Pick a number: 3
>>> resp + 1
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: Can't convert 'int' object to str implicitly
>>>
```

Your Answer:

>>>resp = input('Pick a number:')

Pick a number: 3

>>int(resp)+1

4

- Here the error was 'resp' is a string and to increment it by +1 is not possible since 'str+int' is not possible so we perform type conversion by doing int(resp) then add +1 to make the code work.

Input:

```
resp = input('Pick a number: ')
resp = int(resp) + 1
print(resp)
```

Output:

Pick a number: 3

4

input() returns a string and '+' is not defined for integers and strings so Python attempts to convert the integer 1 to a string but fails. Instead, convert 'resp' to an integer and then add, e.g.

```
resp = input('Pick a number: ')
int(resp) + 1
```

Question 3

2 / 2 pts

Python does **not** include a case/switch statement

- (•) True

  That's right! There's no case/switch statement in Python. Later on, we'll see how to emulate a case/switch statement.

- ( ) False

Question 4

2 / 2 pts

Every file and every function should begin with a

docstring

Question 5

2 / 2 pts

What should you do **first** when you're not sure what Python will do with some expression?

- [ ] Use it anyway. Users will report a bug if it's not right.
- [ ] Give up
- [ ] Ask a friend
- [x] Type it into Python and try it

Question 6

0 / 0 pts

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source."

- (•) True
- ( ) False

Quiz Score: **10** out of 10

Quiz 02

Due Feb 5 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	1,290 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Feb 5 at 5:54pm
This attempt took 1,290 minutes.

Question 1

2 / 2 pts

How can Python programmers avoid adding instance variables caused by typing mistakes?

Your Answer:

We can use __slots__ to avoid adding instance variables caused by typing mistakes.

Use the __slots__ attribute to explicitly specify the attributes for the class.

Question 2

2 / 2 pts

I want to define a class 'Quiz' that is initialized with a single parameter x. Python didn't complain when I defined the class but I get an error **"TypeError: object() takes no parameters"** when I try to create an instance of class Quiz as show below.

```
class Quiz:
    def init(self, x):
        self.x = x
```

```
>>> q = Quiz(10)

TypeError: object() takes no parameters


>>> question = Quiz(3)

TypeError: object() takes no parameters
```

Fix the code so I can create and initialize instances of class Quiz properly.

Your Answer:

```
class Quiz:
    def __init__(self, x):
        self.x = x
```

Question 3

2 / 2 pts

Explain how encapsulation can help to improve your code. Why should you use encapsulation? What can go wrong if you don't?

Your Answer:

- Encapsulation helps packing an information and the methods that work on that information in a same class.
- Encapsulation helps in hiding the data instances so that user can not use it indirectly in any other method. It also helps in code re-usability and adding additional features in future.
- It may happen that you made your project but in future if you want to add some extra features or other developer handles your code it becomes very difficult to understand the code if it's not encapsulated.

Question 4

2 / 2 pts

Describe a situation where you might raise an exception. Write the code to raise a ValueError exception to warn users about an invalid value, x, which is negative, but should be >= 0

Your Answer:

```
x:int = int(input("Enter the Number : "))
if x<0:
    raise ValueError("X is an Invalid and a Negative integer")
```

if x < 0:

raise ValueError('x has value' + str(x) + 'but must be >= 0')

Question 5

2 / 2 pts

What is the output of the following code:

```
def raise_exception():
    raise ValueError
    print("leaving raise_exception()")

def inner():
    raise_exception()
    print("leaving inner()")

def outer():
    inner()
    print("leaving outer()")

def way_out():
    try:
        outer()
    except ValueError:
```

```
        print("way_out(): caught a ValueError")
    print("leaving way_out()")

way_out()
```

Your Answer:

way_out(): caught a ValueError
leaving way_out()

way_out(): caught a ValueError leaving way_out()

way_out() calls outer().
outer() calls inner().
inner() calls raise_exception().
raise_exception() raises a ValueError exception that is not caught in raise_exception() so Python checks in inner(). Inner() does not catch the exception so Python pops the stack and returns to outer(). Outer() does not catch the exception so control returns to way_out(). way_out() catches the exception, prints the "caught a ValueError" message then executes the next line which prints the "leaving way_out()" message.

Quiz Score: **10** out of 10

Quiz 03

Due Feb 12 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	859 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Feb 12 at 5:39pm
This attempt took 859 minutes.

Question 1 — 2 / 2 pts

Describe Test Driven Development. Which comes first, the code or the tests?

Your Answer:

The primary goal of Test driven development is to make code clear and simple. It encourages the developer to write test cases for each functionality and only if the automated test cases fail then the user to need to change the code which also lowers the duplication of code.

Test comes first and then code.

Question 2 — 2 / 2 pts

What are the advantages and disadvantages of debugging with print statements?

Your Answer:

Advantages: Anyone can perform it. It is simple to use.

Disadvantages: To remove print statements may be difficult at times because they may be used at many places and it is necessary to remove it after debugging.

Question 3

2 / 2 pts

How do breakpoints help with debugging?

Your Answer:

Breakpoints are a pause which can be set manually by the developer or debugger to test the program with certain intervals and check if the program works as expected. It also helps in detecting and fixing bugs in the program.

Question 4

2 / 2 pts

What is the difference between step in and step over debugger commands?

Your Answer:

Step in: Step in basically executes line by line. If the line contains a function it will debug the function line by line.

Step over: It is an action to step over the given line. It executes the function and then directly returns the output without executing the function line by line.

Question 5

2 / 2 pts

Describe the divide and conquer strategy for debugging.

Your Answer:

It is a repetitive process of dividing your code in halves until you find the bug. For example, if there's more than a thousand lines of code so it'll be difficult to find the bug in it so using Divide and conquer we can do is divide the code in half, then find which half has bugs and the repeat the same process until you find the bug.

Quiz Score: **10** out of 10

Quiz 04

Due Feb 19 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides or any other source. You will receive a score of 0 if you copy the prose from the source.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	461 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Feb 19 at 7:17pm
This attempt took 461 minutes.

Question 1 — 2 / 2 pts

What is the value of **list(range(5, 15, 3))**?

- [5, 10, 15]
- [5, 8, 11, 14] — Correct!
- [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 3]
- [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

Question 2 — 2 / 2 pts

What values are printed by the following code?

```
for i in range(5):
    if i == 4:
        break
    elif i == 2:
        continue
    print(i)

print('done')
```

- 0, 1, 2, 3, 4, 5, done
- 1, 2, 3, done
- 0, 1, 2, 3, 4, done
- 0, 1, 3, done

Correct!

Question 3

2 / 2 pts

What's wrong with this code segment? Please fix it to print

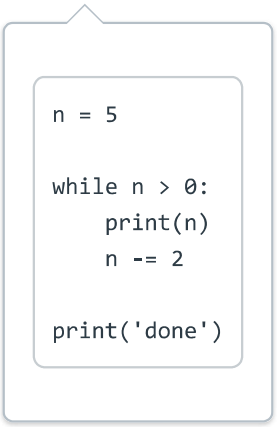
5
3
1
done

```
#####
## buggy code ##
n = 5

while n != 0:
     print(n)
     i -= 2

print('done')
```

Your Answer:

n = 5
while n > 0:

print(n)
n -= 2
print('done')



Question 4

2 / 2 pts

What is the output from the following code segment?

```
for i in range(0,4,2):
    for j in range(2):
        print(i, j)
```

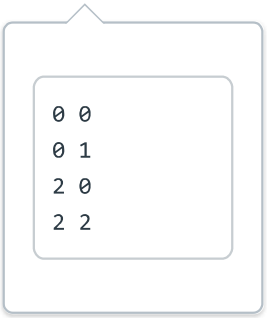
Your Answer:

0 0

0 1

2 0

2 1



Question 5

2 / 2 pts

Describe the characteristics of situations when generators are a good solution?

Your Answer:

Anytime we need sequence we can use generators. Instead of making and storing big data lists in advance it'll be much more efficient to use generator. Generators are used for sudden events, they get active on request call and stay active until requested and then go to rest again.

Quiz Score: **10** out of 10

Quiz 05

Due Feb 26 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	77 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Feb 26 at 5:43pm
This attempt took 77 minutes.

Question 1 — 2 / 2 pts

In what order are foo() and bar() called in object.foo().bar()?

Correct!

- (•) foo() first, then bar()
- ( ) bar() first, then foo()
- ( ) Neither: that's invalid syntax

Question 2 — 2 / 2 pts

Write three separate code blocks to print each character in the string 'hello':

1. using a for loop, range, and a slice ([])
2. using a for loop without range or slice ([])
3. using a while loop and a slice ([])

(Note: a for loop is probably a better solution for this problem but here you're restricted to range, while, and slices)

Hint: len('Hello') == 5

Here's a solution using a for loop:

```
for ch in 'hello':
     print ch
```

Your Answer:

1.

```
s="hello"
for i in range(len(s)) :
        print(s[i])
```

2.

```
for i in "hello":
           print(i)
```

3.

```
i=0
s="hello"
while i < len(s):
       print(s[i])
       i+=1
```

```
s = "hello"

# for loop, range, and index
for i in range(len(s)):
    print(s[i])

# for loop, without range, without index
for c in s:
    print(c)

# while loop and index
i = 0
while i < len(s):
    print s[i]
    i += 1
```

Question 3

2 / 2 pts

Describe two different Python string methods to determine if a string 'abc' occurs anywhere inside the string 'xxxabcxxx'.

Your Answer:

Two different Python string methods to determine if a string 'abc' occurs anywhere inside the string 'xxxabcxxx' are : **find()** , **index() .** Both methods return an index for a specific sub-string if found in a given/mentioned string, so using these methods we can know if the sub-string is present in the string or not.

s="xxxabcxxx"
print(s.find("abc"))
print(s.index("abc"))

Output:

3

3

```
'xxxabcxxx'.find('abc')
```

```
'abc' in 'xxxabcxxx'
```

in/not in is another way to find the occurrence of a substring in a string.

Question 4

2 / 2 pts

What is the effect of using negative numbers to extract a slice of a string? E.g.

'hello'[-3:] or 'hello'[:-4]

Your Answer:

Negative index in the the string work from the end in a reverse manner. For example, if the index is -1 it'll print the last element of the string.

print('hello'[-3:]) will print "llo", it prints all the characters from the third last index.

print('hello'[:-4]) will print all characters from starting to 4th last character which results in printing "h" for the given statement.

Negative indices work from the end (or beginning) of the string.
s[-i] == s[len(s) - i]

Question 5

2 / 2 pts

Why is it important to follow coding style guidelines?

Your Answer:

It is important to follow coding style guidelines because it improves code readability and maintainability. Coding style guidelines make easier for user to read and understand the code since it'll be written in an organized way.

Coding standard guidelines help to make code more readable and maintainable. And you'll lose points on your assignments if you don't follow coding standard guidelines.

Quiz Score: **10** out of 10

Quiz 06

Due Mar 4 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	60 minutes	10 out of 10

ⓘ Correct answers are hidden.

Score for this quiz: **10** out of 10
Submitted Mar 4 at 9:49pm
This attempt took 60 minutes.

Question 1 — 2 / 2 pts

Describe the four types of Python containers. How are they different from each other?

Your Answer:

Python Containers :

1. Lists: Lists have arbitrary values with mutable elements and the elements are ordered.

2. Tuples: Tuples are also ordered like lists but are the elements of the tuple are immutable that means the elements once defined can not be changed.

3. Dictionaries: Dictionaries have a unique key for every value, it contains a pair of key, value. Also, the dictionaries are mutable, unlike tuples.

4. Sets: Sets are unordered and have all unique elements in them and are also mutable.

Lists are ordered and mutable with elements of any type.

Tuples are ordered, but not mutable with elements of any type.

Dicts map hashable keys to values. Dicts are mutable.

Sets contain distinct hashable values and are mutable.

Question 2

2 / 2 pts

Explain the difference between list.append() and list.extend(). Include an example in your answer.

Your Answer:

1. list.append(): list.append() adds new values at the end of the list.

2. list.extend(): list.extend() concatenates two lists.

For example:

lst=['a', 'b', 'c']

lst.append(['1', '2'])

Output: ['a', 'b', 'c', ['1', '2']]

lst=['a', 'b', 'c']

lst.extend(['d'])

Output: ['a', 'b', 'c', 'd']

list.append(value) appends the value a new element at the end of the list.

list.extend(sequence) concatenates list and the sequence.

Question 3

2 / 2 pts

Write a code fragment that shows how to emulate the behavior of list.remove(value) using only list.pop() and list.index()

Your Answer:

```
def lst_remove(i,lst):
        if i in lst:
            a= lst.index(i)
             return(lst.pop(a))


lst_remove(1,[1,3,2,4])
```

```
def my_remove(t, l):
    if t in l:
       l.pop(l.index(t))
```

Question 4

2 / 2 pts

What is the difference between sorted(list) and list.sort()? Show examples of both.

Your Answer:

sorted(list) doesn't make any changes in the original list and returns a copy of the list in a new list with the sorted elements.

list.sort() modifies the original list and sorts it.

lst=[6,4,5]

sorted(lst)

Output: [4,5,6]

lst.sort()

Output: lst=[4,5,6]

sorted(list) returns a sorted **copy** of the list

list.sort() sorts the list in place

Example from slides:

```
lst1 = [3, 1, 2]
sorted(lst1) returns a new list [1, 2, 3].  lst1 is not changed

lst1.sorted() sorts lst1 in place, changing the list so
lst1 == [1, 2, 3]
```

Question 5

2 / 2 pts

Consider the code fragment:

```
x = [1, 2, 3]
y = [x, x]
x[0] = 4
```

What is the value of y? Why?

Your Answer:

y=[[4,2,3],[4,2,3]]

Because x[0] = 4 changes the value of x to [4,2,3] from [1,2,3] and y=[x,x] thus y contains x which changes the value of y too

```
[[4, 2, 3], [4, 2, 3]]
```

Y consists of two instances of x so changing x also changes y.

Quiz Score: **10** out of 10

Quiz 07

Due Mar 11 at 11:59pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	1,159 minutes	10 out of 10

Correct answers are hidden.

Score for this quiz: **10** out of 10
Submitted Mar 11 at 6:44pm
This attempt took 1,159 minutes.

Question 1

2 / 2 pts

A Temporary Employment Agency hires and fires employees frequently using the strategy where the person hired most recently is the first person fired. You are responsible for writing a function to track when employees are hired and identifying the next person to be fired. Which builtin Python container would you use (list, tuple, dict, or set)? Which methods of that container would you use?

Your Answer:

To perform this operation we'll need to implement STACK which works on the basis of LIFO(Last In First Out) and to perform this operation we'll need to implement it using List.

And to perform this operation using list I would use list.append() to add the employee name at last and to remove the name from the beginning I'll

use list.pop().

Use a stack (LIFO) which can be implemented with a list, and either

list.insert(0, value), and list.pop(0) # insert at the beginning, pop from the beginning

or

list.append(value) and list.pop() # append at the end and pop from the end

Question 2

2 / 2 pts

You are asked to write a program that reads an arbitrary file and identify the 10 most frequently used words. Which Python container would you choose and why?

Your Answer:

The python container we can use is the function Counter() because it can count the occurrences of letters or words in a given string.

defaultdict(int) or Counter()

Question 3

2 / 2 pts

Explain how memoization works and how it can improve performance for some algorithms. Under what conditions does memoization help? Under what conditions, does it **not** offer much benefit?

Your Answer:

Memoization is the process to avoid recalculation and thus it increases the speed of calculation by saving time not recalculating it. It works in a way that it stores the result of a function and saves time not recalculating it if the calculation is to be done on the same elements.

Memoization can done during some programs where the function contains repetitive calls, for example: Factorial.

Memoization doesn't offer much benefit if the function doesn't have repetitive calls or has to compute a value less number of times.

Memoization stores intermediate results to avoid recalculation. This is very helpful for problems where the same request may be made many times, e.g. fibonacci, factorial, etc. Memoization does not help if the solution computes a value a small number of times.

Question 4

2 / 2 pts

You need to write a function that manages a customer waiting list. As the customer enters the store, she adds her name to the waiting list. When an employee becomes available, the employee identifies the customer who has been waiting the longest and then removes that customer's name from the list, and helps the customer. Which builtin Python data structure is most appropriate (list, tuple, dict, or set)? Which methods would you use?

Your Answer:

Here we'll need to implement a Queue, thus the built-in Python data structure which will be most appropriate will be using List.

In List to perform queue operations we can use list.insert() or list .append() to add the customer's name and the function such as list.pop() can be used to remove the name of the customer from the waiting list.

Use a queue (FIFO) which can be implemented with a list, and either

list.insert(0, value), and list.pop() # insert at the beginning, pop from the end

or

list.append(value) and list.pop(0) # append at the end and pop from the beginning

Question 5

2 / 2 pts

Describe a situation where lists are not appropriate but tuples are a good match.

Your Answer:

When you need to store a large dataset which shouldn't be changed then at that time you can use Tuples over lists because Tuples are immutable and the values can't be changed also if tuples are used the other developers will come to know that those values shouldn't be changed. When we need to group multiple values in a single element or need to swap values. Tuples are kind of lists that are immutable.

Tuples can be used as the key in a dictionary but lists can't be used a the key for a dictionary

Quiz Score: **10** out of 10

Quiz 08

Due Mar 25 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides, the web, or any other source. You will receive a score of 0 if you copy the prose from any source.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	108 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Mar 25 at 6:53pm
This attempt took 108 minutes.

Question 1 — 2 / 2 pts

Given

```
def foo(x, use_max = True, *values):
    return x + (min(values) if use_max else max(values))
```

```
>>> foo(1, 2, 3, 4) # call foo()
```

1. What is the value of the use_max?

2. What is the value of 'values'?

3. What value is returned from the function call? Why?

Your Answer:

1. use_max = 2

2. values = (3,4)

3. Here if condition goes True since use_max=True, thus the function will return "x + (min(values)"

which is 1 + min(3,4) = 1 + 3 = 4

1. use_max is assigned the value of the second parameter so use_max == 2

2. values == (3, 4)

3. use_max evaluates to True so the expression becomes 1 + min(3, 4) == 4

Question 2

2 / 2 pts

Python provides three different ways to import modules:

import module_name as alias

from module_name import objects

from module_name import *

Why should you avoid "from module_name import *"?

Your Answer:

"from module_name import *" can raise many conflicts since it imports all the functions present in the module so if there's an existing function name that matches any of your variable or function names then the conflict will occur.

import * may redefine important function definitions with potentially dangerous alternatives that can cause unexpected results.

Question 3

2 / 2 pts

Write a function that demonstrates a good use of using keyword parameters. Don't use the examples from the lecture.

Your Answer:

```
def students(**kwargs = Any):
        for key,value in kwargs.items():
            print(f"{key} studies in {value}")
students(Rohan Ratwani = Stevens Institute of Technology)
```

Output:

Rohan ratwani studies in Stevens Institute of Technology

Example from the lecture:

```
def keywords(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}={value}")

keywords(name="jim", age="ancient", happy=True)
```

Question 4

2 / 2 pts

You are starting a project with several developers to write a large new Python program. Explain how you can use modules to organize the code for your project.

Say your project has file1.py, file2.py, file3.py, and file4.py. File2.py needs class foo from file3.py and class bar from file4.py. Write the Python code that you would include in file2.py to make foo and bar available.

Your Answer:

Using python modules we can import classes and function from the other existing files which encourages code reusability and also saves a lot of time.

To make foo and bar available in file2.py we need to write at the starting of file2.py:

from file3 import foo

from file4 import bar

file2.py should include the following lines:

from file3 import foo

from file4 import bar

Question 5

2 / 2 pts

How can pypi.org help to make Python programmers more productive?

Your Answer:

pypi.org is a site for people in the python community which consists of lakhs of projects and other lakhs of packages that can be searched and installed directly. This may help the user to learn easily and also get ready modules and projects which the user will not need to start from scratch.

Rather than solving all problems from scratch, Pythonistas can look to PyPI.org to see if existing solution already exist.

Quiz Score: **10** out of 10

Quiz 09

Due Apr 1 at 10pm **Points** 10 **Questions** 6 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	84 minutes	10 out of 10

Score for this quiz: **10** out of 10
Submitted Apr 1 at 7:57pm
This attempt took 84 minutes.

Question 1 — 2 / 2 pts

You are responsible for measuring the performance of a critical Python production application. Specifically, you need to collect and log the date and time that each of three critical functions in the application is called. Your team has a logging module that will store the data, but how can you use decorators to cause the logging to occur without changing the implementation of the existing functions?

NOTE: You should **not** write the code for the decorator, just describe how decorators can be used to help to solve this problem.

Your Answer:

First, we can define the decorator by (@functionname), @log_it the use of this function is that it determines the current date and the time and then calls the logging module to log the function call. We can update the code by writing.

@log_it

def func1()

    pass

@log_it

def func2()

    pass

@log_it

def func3()

    pass

Define a decorator, @log_it, that determines the current date and time and then calls the logging module to log the function call. Update the source to wrap the functions:

```
@log_it
def func1():
    pass

@log_it
def func2():
    pass

@log_it
def func3():
    pass
```

Question 2

2 / 2 pts

Compare and contrast static methods and non-static methods in Python classes.

Your Answer:

The main difference between a static and a non static class is that a static class doesn't include 'self' as a parameter because the method applies to

a class definition rather than one specific instance of a class.

Whereas, on the contrary non-static methods have a 'self' parameter because the method is associated with a specific instance.

Non-static methods always include 'self' as the first parameter in the method. 'self' is used to access and/or modify the specific instance of the class.

Static methods don't include a 'self' parameter because the method applies to the class definition rather than a specific instance of the class.

Question 3

2 / 2 pts

Compare and contrast ***instance*** attributes and ***class*** attributes in Python classes.

Your Answer:

Instance attributes can be referred to as the replica of every instance in a class. Every instance has a distinct role in a class.

Whereas, a class is common across all the instances of a class and a single instance can be shared across every other instance of the class.

Each instance of a class C includes a copy of every instance attribute so changing the instance attribute in one instance of the class does not impact the value in any other instance of the class.

Class attributes are shared across all instances of the class so changing the class attribute in any one instance changes all of them.

Question 4

2 / 2 pts

Describe Duck Typing in Python. How does Python know to apply '+' appropriately with different types? E.g.

```
x = 1 + 2

y = "hello " + "world"
```

Your Answer:

Duck typing is a dynamic typing method that allows objects of different types to respond to the same methods. Duck typing is used to check the presence of the given method. Python decide on the appropriate class method based on the type of object at run time.

Python uses Duck Typing to support polymorphism without inheritance. Python checks the types of both the left and right operand to see if both are the same type and applies the magic method if that type defines the major method associated with the operand, e.g. '+' maps to '__add__'.

Question 5

2 / 2 pts

Describe the naming conventions used by Pythonistas to support public, protected, and private attributes and methods in Python code

Your Answer:

name (no _) tells the reader that name is public
_name (single _) tells the reader that _name is protected
__name (double _) tells the reader that __name is private and shouldn't be used

name (no _) tells the reader that name is public

_name (single _) tells the reader that _name is protected

__name (double _) tells the reader that __name is private and shouldn't be used

Question 6

0 / 0 pts

"I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination. I further pledge that I have not copied any material from a book, article, the Internet or any other source except where I have expressly cited the source."

Correct!

- (•) True
- ( ) False

Quiz Score: **10** out of 10

Quiz 10

Due Apr 8 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	52 minutes	10 out of 10

ⓘ Correct answers are no longer available.

Score for this quiz: **10** out of 10
Submitted Apr 8 at 9:05pm
This attempt took 52 minutes.

Question 1 — 2 / 2 pts

Select all of the correct statements

- [x] Refactoring is an important tool for software development
- [x] Refactoring can help to improve code readability
- [ ] Only code should be refactored, not design
- [x] Test Driven Development makes refactoring easier
- [x] Refactoring helps you to find bugs

Question 2

2 / 2 pts

True or False and why?: Configuration Management is only important for big projects so I don't need to worry about it. Why?

Your Answer:

False, Whether the business or project is small but the configuration management is must because it helps building and identifying the configuration of a project and helps to keep track for controlling changes do the needful changes in the configurations.

Configuration management is important for all projects, both large and small.

A good CM solution has many advantages that are important to all teams including:

- each access to all versions of important documents, e.g. code

- CM helps to facilitate collaboration across teams by allowing different people to work on the same files simultaneous

- CM provides backup across devices

Question 3

2 / 2 pts

What information should we track in configuration management for your current project?

- [x] Source code files
- [x] Code Documentation

- [x] Architecture and Design documents
- [x] Issues and design decisions

Question 4

2 / 2 pts

You've been assigned to work on a new release of a development project with three other developers. The project has a core set of classes, developed in the previous release, that are shared by all developers and each developer is responsible for implementing independent features for the next release. The project GitHub repository currently has only a "master" branch. Describe an optimal branching solution for the upcoming release. Will all four developers work and commit directly to the "master" branch? Will you create one or more sub-branches? What are the benefits of your solution?

Your Answer:

I would suggest to create multiple branches for the each team member and if needed then they can merge it into master. But forming different branch will help the team members to work on their given task independently without any conflicts.

One good solution is to fork a new "dev" branch from the "master" branch where the developers will push their changes. It's best to have everyone working in the same branch so the branches don't get too far out of sync and any integration problems can be identified and fixed quickly.

Question 5

2 / 2 pts

How is the merge done by a pull request different from the merge that is done when pushing or pulling code from a branch in GitHub? Hint: who is involved in a pull request vs a push/pull.

Your Answer:

While doing push or pull the developer is involved and the changes are done directly to the the branch and if we use pull request then the request can be seen by the team and other developers review the changes and approve the request to merge.

Anyone can merge changes into a branch. Pull requests are a request from contributors for the project leaders to review the changes and either to approve or reject those changes.

Quiz Score: **10** out of 10

Quiz 12

Due Apr 22 at 10pm **Points** 10 **Questions** 5 **Time Limit** None

Instructions

Answer the following questions in your own words. Do NOT simply cut and paste the information from the slides. You will receive a score of 0 if you copy the prose from the slides.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	219 minutes	10 out of 10

ⓘ Correct answers are no longer available.

Score for this quiz: **10** out of 10
Submitted Apr 17 at 8:58pm
This attempt took 219 minutes.

Question 1 — 2 / 2 pts

How are dynamic web pages different from static web pages?

Your Answer:

In dynamic web pages the content of the the web page can get updated dynamically with interval of time or show the content respective to the user. Whereas, in static web pages the content can not be changed unless and until the developer changes it manually.

Dynamic web pages build HTML pages dynamically when requested and in static the HTML pages are defined.

Dynamic web page uses database where as static web pages may not need to use any database.

Static web pages are static: the content doesn't change. Dynamic web pages may change because they are created on the fly and may access external data sources such as databases.

Question 2

2 / 2 pts

Describe how Flask uses Jinja2 to support dynamic web pages.

Your Answer:

In Flask Jinja2 provides a bridge between dynamically generated HTML files and python programs.Also if the user needs the same layout all over different webpages the template inheritance can be the most useful thing since it has one base template which contains headers and footers to maintain layout.

Flask uses Jinja2 to combine data provided the Python program to populate fields in the HTML template files

Question 3

2 / 2 pts

Python **decorators:**

- ○ Do nothing except make Python code hard to read
- ◉ augment, without changing, the behavior of a function or class
- ○ decorate cakes when not being used in Python programs

augment and change the behavior of functions and classes

Question 4

2 / 2 pts

The HTML files in your Flask solution can be stored anywhere relative to your .py files

True

False

The .HTML templates must be stored in the ./templates directory of your project

Question 5

2 / 2 pts

Flask is a closed system that cannot be extended to add new features.

True

False

Quiz Score: **10** out of 10