Course Assessment Test

Course Title	Introduction to Python	Date	
Name		Dept	

1) *This assessment test is to be given out before course commencement. Answers are to be filled in column entitled "Pre-Course Answer"*
2) *At the end of the course, the same assessment sheet is to be given out where answers are to be filled in column entitled "Post-Course Answer". Instructor will then share the answers and participants need to total the score in both "Pre" and "Post" columns through self-marking.*
3) *Assessment sheets will be collected for filling.*

No	Question	Pre-Course Answer	Post-Course Answer
1	Identify which of the following statements about the Python programming language is correct i. Python programs are placed in text files called scripts or modules ii. The reference implementation of the Python interpreter is written in Java (known as Jython) iii. The Python interpreter can run Python code as a script or module or as statements in an interactive session in the REPL shell. iv. The Python interpreter can perform code optimization for modules by compiling to an intermediate format known as bytecode a) Statements i, ii and iii b) Statements i, ii and iv c) Statements i, iii and iv d) All statements are correct		C
2	Identify the statement that correctly creates a constant (i.e. a variable whose value cannot be changed after its initial assignment) a) `const valOfPi = 3.14` b) `fixed myConstVal = 25` c) `FILE_SIZE_LIMIT = 2000` d) It is not possible to declare a constant in Python		D
3	Identify which of the following `print` statements will execute correctly without an error: i) `name = 'Pete'` `print('Hello %s' % name)`		D

ii)

```
price = 22.6784
print('This cup costs %.2f dollars' % price)
```

iii)

```
name = 'John'
age = 20
print("Hello I'm {}, my age is {}".format(name, age))
```

iv)

```
name = 'Elizabeth'
print(f'Hello {name}!')
```

a) Statements i, ii and iii
b) Statements i, ii and iv
c) Statements i, iii and iv
d) All statements will execute correctly

4 Consider the snippet of the code below

```
def greet(name):
    print(f"Hi {name}")


message = greet("superman")
print (message)
```

What is the output when it is executed?

A

a)

```
Hi Superman
None
```

b)

	`` Hi Superman Hi Superman ``` c) ``` Hi Superman Null ``` d) ``` Hi Superman Traceback ReturnError -> function does not return any assignable value ```		
5	Consider the initial content of the list below: `numbers = [1, 2, 3, 4]` Which of the following statements will produce the following altered content for the same list: `[1, 2, 4]` i) ``` del numbers[2] ``` ii) ``` numbers.pop(3) ``` iii) ``` numbers.remove(3) ``` iv)		B

	`` numbers.del(2) ``` a) Statements i and ii b) Statements i and iii c) Statements ii and iii d) Statements ii and iv		
6	Consider the list below: `colors = ['red', 'orange', 'yellow', 'green']` Which of the following statements below will print out the list in reverse order, i.e. `['green', 'yellow', 'orange', 'red']` a) ``` print(colors[-1::]) ``` b) ``` print(colors[::-1]) ``` c) ``` print(colors[0:3:4]) ``` d) ``` print(colors[0:4:3]) ```		B
7	All of the following snippets of code will execute correctly EXCEPT for: a) ``` ```		D

```
def greet(name, message='Hi'):
    return f"{message} {name}"
greeting = greet('John')
print(greeting)
```

b)

```
def greet(name='there', message='Hi'):
    return f"{message} {name}"
greeting = greet()
print(greeting)
```

c)

```
def greet(name='there', message='Hi'):
    return f"{message} {name}"
greeting = greet(message='Hello')
print(greeting)
```

d)

```
def get_net_price(price, tax=0.07, discount=0.05):
    return price * (1 + tax - discount)
net_price = get_net_price(100, tax=0.08, 0.06)
```

8 Consider the snippet of code below:

```
bonuses = [100, 200, 300]
new_bonuses = []
for bonus in bonuses:
    new_bonuses.append(bonus*2)
print(new_bonuses)
```

Which of the following snippets of code listed below will accomplish the same functionality as the previous one?

C

a)

```
bonuses = [100, 200, 300]
print(map(list(func bonus: bonus*2, bonuses)))
```

b)

```
bonuses = [100, 200, 300]
print (list(map(func bonus *= 2 , bonuses)))
```

c)

```
bonuses = [100, 200, 300]
print(list(map(lambda bonus: bonus*2, bonuses)))
```

d)

```
bonuses = [100, 200, 300]
print(map(list(bonus => bonus*2, bonuses)))
```

9 Which of the following unpacking related statements will execute correctly without any error thrown by the interpreter?

i)

```
x, y, z = 10, 20, 30, 40
```

ii)

```
x, y, _ = 10, 20, 30
```

iii)

```
a, b, *c = ['cat','dog','mouse','rat']
```

iv)

```
a, b, *c, *d = [1,2,3,4,5,6,7,8]
```

C

	a) Statements i and ii b) Statements i, ii and iii c) Statements ii and iii d) Statements ii, iii and iv		
10	Identify the correct locations where Python will search for a module when the `import` statement is encountered in a script: i. The `modules` subfolder in the main installation directory of Python ii. The current folder holding the executing script as well as all its parent folders in the hierarchy of folders iii. A list of folders specified in the PYTHONPATH environment variable, if this is set iv. An installation-dependent list of folders configured during the installation of Python (such as `\lib\site-packages`) a) Locations i and ii b) Locations i and iii c) Locations ii and iii d) Locations iii and iv		D
Total			