Week 11 - Practice Theory

Problem 1

Question

What function `f()` will return? [MCQ]

```
def f():
    try:
        return 1
    except:
        return 2
    finally:
        return 3
    return 4
```

a) 1

b) 2

c) 3

d) 4

Answer

(c)

Solution

`finally` block is always executed so, the return value of `finally` block will override the previous return value in function then the output will be 3.

Problem 2

Question

What is the output of the following code? Select the most appropriate answer. [MCQ]

```
L = [0, 1, 2, 3]
print(L[4])
```

(a) `3`

(b) `ValueError`

(c) `IndexError`

(d) None of these

Answer

(c)

Solution

In `print(L[4])` statement, index 4 is not exist in list `L` then `IndexError` will occur.

Problem 3

Question

Match the `Error Name` with the correct `Description`. [MCQ]

Name	Description
1. `IndexError`	A. Raised when a variable is not found in the local or global scope.
2. `NameError`	B. Raised when a function or operation is applied to an object of an incorrect type.
3. `KeyError`	C. Raised when the index of a sequence is out of range.
4. `SyntaxError`	D. Raised by the parser when a syntax error is encountered.
5. `TypeError`	E. Raised when a key is not found in a dictionary.
6. `IndentationError`	F. Raised when there is an incorrect indentation.
7. `ValueError`	G. Raised when the second operand of a division or module operation is zero.
8. `ZeroDivisionError`	H. Raised when a function gets an argument of correct type but improper value.

(a) 1-E, 2-C, 3-A, 4-D, 5-F, 6-B, 7-H, 8-G

(b) 1-A, 2-C, 3-E, 4-F, 5-D, 6-B, 7-H, 8-G

(c) 1-C, 2-A, 3-E, 4-D, 5-B, 6-F, 7-H, 8-G

(d) 1-F, 2-C, 3-E, 4-A, 5-G, 6-B, 7-H, 8-D

Answer

(c)

Solution

Self explanatory

Problem 4

Question

```
l = []
for i in range(10):
    if(i % 2 == 0):
        l.append(i)
    else:
        l.append("Odd")
```

Which of the following codes are equivalent to the above code snippet? [MSQ]

(a)

```
l = [i if i % 2 == 0: else: "Odd" for i in range(10)]
```

(b)

```
l = [i if i % 2 == 0 else "Odd" for i in range(10)]
```

(c)

```
l = ["Odd" if i % 2 != 0 else i for i in range(10)]
```

(d)

```
l = [l.append(i) if i % 2 == 0 else "Odd" for i in range(10)]
```

Answer

(b), (c)

Solution

Option (b) and (c) are the correct ways to use if-else inside list comprehension, so these options are equivalent to the given code

Problem 5

Question

```
def fun(n):
    if n % 2 == 0:
        return lambda x : x ** n
    else:
        return lambda x : n ** x

L = [fun(i)(2) for i in range(1,10)]
```

What will be the value of `L` after executing the above code snippet? [MCQ]

(a)

```
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

(b)

```
[81, 64, 49, 36, 25, 16, 9, 4, 1]
```

(c)

```
[1, 4, 9, 16, 25, 36, 49, 256, 81]
```

(d)

```
[1, 4, 9, 16, 25, 64, 49, 256, 81]
```

Answer

(d)

Solution

`L = [fun(i)(2) for i in range(1,10)]` . Here `i` will be passed as the argument `n` to the function `fun`. So, `fun(i)` returns a `lambda` function. In `fun(i)(2)` , the argument `2` will be passed as x to the `lambda` function returned by `fun(i)`. In short, `fun(n)(x)` is equivalent to `fun(i)(2)` . Therefore `fun(i)(2)` returns 2^i or i^2 based on whether `i` is divisible by `2` or not.

Problem 6

Question

The variable `fruit_iter` is an iterator object on a list `fruit_list`. Which of the following statement is True? [MCQ]

```
fruit_list = ["apple", "mango", "orange"]
fruit_iter= iter(fruit_list)
```

(a) The iterator `fruit_iter` can access elements from `left to right` only

(b) The iterator `fruit_iter` can access elements from `right to left` only

(c) The iterator `fruit_iter` can access elements in both directions `left to right` and `right to left`

(d) None of these

Answer

(a)

Solution

For performing an iteration operation using the iterator object, the python had two special methods: `iter()` and `next()`. The `iter()` function returns an iterator object of the given argument (a collection). This object has a `next()` method to access elements in the collection one at a time. It traverses from left to right. When there are no more elements, `next()` raises a `StopIteration` exception which tells the for loop to terminate.

Problem 7

Question

```
prog = {"a": "PHP", "b":"JAVA", "c":"PYTHON", "d":"C++"}
for i in enumerate(prog,1):
    print(i)
```

What will be the output of the above code-snippet?

(a)

```
(0, 'a')
(1, 'b')
(2, 'c')
(3, 'd')
```

(b)

```
(1, 'a')
(2, 'b')
(3, 'c')
(4, 'd')
```

(c)

```
a
b
c
d
```

(d)

```
PHP
JAVA
PYTHON
C++
```

(e)

```
[1, 'a']
[2, 'b']
[3, 'c']
[4, 'd']
```

Answer

(b)

Solution

`for i in enumerate(prog,1):` here in each iteration, `enumerate` function returns a tuple to `i` where the first element will be the count number (starting from 1) and the second element will be the key of `prog` dictionary. Hence, the correct option is (b).

Problem 8

Question

For what values of `a`, `d` and `n`, the following options the below generator function generates below numbers. [MCQ]

```
1,3,5,7,9,11,13,15
```

Code

```
def arithmetic_progression_generator(a, d, n):
    count = 1
    while count <= n:
        yield a
        a += d
        count += 1
```

(a) `a = 0, d = 2, n = 10`

(b) `a = 0, d = 2, n = 8`

(c) `a = 1, d = 2, n = 10`

(d) `a = 1, d = 2, n = 8`

Answer

(d)

Solution

The function yields `a` as the first element of the sequence. Next, it yields `a + d` which is 3. The next element yielded is `a + 2 * d` which is 5. When `n = 8`, all the numbers in the sequence are generated, which is `1,3,5,7,9,11,13,15`.

Problem 9

Question

```
a = [10, 20, 30, 40, 50]
b = [1, 2, 3, 4, 5, 6, 7]
L = list(map(lambda n, m: m * n, a, b))
```

What will be the value of `L` after executing the above code snippet?[MCQ]

(a)

```
[10, 40, 90, 160, 250, 6, 7]
```

(b)

```
[10, 40, 90, 160, 250]
```

(c)

```
[10, 40, 90, 160, 250, 300, 350]
```

(d) `IndexError`

Answer

(b)

Solution

In the above code, `map` function will read elements one by one from both lists `a` and `b` and pass it to the lambda function as `m` and `n`. The lambda function returns `m * n` for each value pair passed to it. The values are stored in the list `L`. If both list `a` and `b` are not of equal size, once items of any list are exhausted while applying map function, further calls to map function halt. Hence, option (b) is correct.

Problem 10

Question

Which of the following ways a list of 10 numbers can be printed using the object `s` of the class `Series`. [MSQ]

```
class Series:
    def __iter__(x, n = 1):
        x.a = 0
        x.n = 1
        return x
    def __next__(x):
        if x.a < 10:
            x.a = x.a + x.n
            return x.a
        else:
            raise StopIteration
s = Series()
```

(a)

```
s_iter = iter(s)
for item in s_iter:
    print(item)
```

(b)

```
for item in s:
    print(item)
```

(c)

```
s_iter = iter(s)
for i in range(10):
    print(next(s))
```

(d)

```
for i in range(10):
    print(next(s))
```

Answer

(a), (b), (c)

Solution

`next()` can be used on the object returned by the function `iter()`. Elements of an `iterable` object can be accessed using `for` loop as well.