QUIZ:- 03

Question 1

Complete

Mark 2.00 out of 2.00

⚑ Flag question

Given the following code:

```
class G():
    def __init__(self, a):
        self.a = a
```

which of the following statements is true?

Select one or more:

☑
```
g = G()
g.a
```
will fail.

☑
```
g = G('a')
g.a
```
will return 'a'

☐ The program will crash because the class definition is syntactically wrong.

Question 2

Complete

Mark 2.00 out of 2.00

⚑ Flag question

Consider the following list in python:

```
x = [1, 3, 4, 7]
```

Which of the following functions produces the output:

```
[4]
```

Select one or more:

☑
```
y = []
for idx in range(len(x)):
    if x[idx]//2==2:
        y.append(x[idx])

print(y)
```

☐ `print([y for y in x if (x > 3) and (x < 7)])`

☑ `print([y for y in x if y%2 == 0])`

Question 3

Complete

Mark 2.00 out of 2.00

⚑ Flag question

Consider the following program

```
def f(*a):
    return [b for b in a if type(b) is str]
```

Which of the following statements is true?

Select one or more:

☑
    `f('a',8,.0)`

    will return

    `['a']`

☑ `f(*['a','8',.0])`

    will return

    `['a','8']`

☑ `f(8,.0)`

    will return

    `[]`

☐ `f(*[a,str,19])`

    will return

    `[a]`

Consider the following list in python:

```
fruit = ['banana','pineapple','apple']
```

Which of the following functions produces the output:

```
['pineapple', 'apple']
```

```
To get the output of a function, you need
to call the function: get_special_fruit(fruit)
```

Select one or more:

☑
```
def get_special_fruit(fruit):
    special_fruit = []
    for f in fruit:
        if 'app' in f:
            special_fruit.append(f)
    return special_fruit
```

☐
```
def get_special_fruit(fruit):
    special_fruit = list()
    while len(special_fruit) == 0:
        special_fruit = fruit[-1]
    return special_fruit
```

☑
```
def get_special_fruit(fruit):
    return [f for f in fruit if 'ap' in f]
```

Which of the following statements about iterators are true?

Select one or more:

☐ Iterators cannot be transformed into lists.

☑ Iterators are faster than generators.

☑ Using iterators can help to reduce the memory footprint of an algorithm.