1. Aggregation & Dependency - Quiz

Q1 of 5

What is the output of the following code snippet?

```
class Customer:
    def __init__(self,name,mobile):
        self.name=name
        self.mobile=mobile
class Mobile:
    def __init__(self,brand):
        self.brand=brand
    def unlock(self,cover):
        print(cover.color)
class Cover:
    def __init__(self):
        self.__color="red"
Customer("Cus1",Mobile("Apple")).mobile.unlock(Cover())
```

- (●) Error ✓
- ( ) red
- ( ) None

Q2 of 5

What is the output of the following code snippet?

```
class Customer:
    def __init__(self,name,mobile):
        self.name=name
        self.mobile=mobile
class Mobile:
    def __init__(self,brand):
        self.brand=brand
    def unlock(self,cover):
        cover.color="yellow"
class Cover:
    def __init__(self):
        self.color="red"
Customer("Cus1",Mobile("Apple")).mobile.unlock(Cover())
print(Cover().color)
```

- ( ) Error
- (●) red ✓
- ( ) yellow

Q3 of 5

How can we access order id from trade object?

Note: A Trade object has a TradeDetail object.(Aggregation)

```
class Trade:
    def __init__(self):
        self.__trade_detail = None

    def get_trade_detail(self):
        return self.__trade_detail

    def set_trade_detail(self, value):
        self.__trade_detail = value

class TradeDetail:
    def __init__(self):
        self.__trade_id = None
        self.__order_id = None

    def get_trade_id(self):
        return self.__trade_id

    def get_order_id(self):
        return self.__order_id

    def set_trade_id(self, value):
        self.__trade_id = value

    def set_order_id(self, value):
        self.__order_id = value
```

a) t=Trade()
t.get_trade_detail()

b) t=Trade()
t.get_trade_detail().get_order_id()

c) t=Trade()
tr=TradeDetail()
t=tr.get_order_id()

d) It cannot be called

- ○ a
- ◉ b ✓
- ○ c

Q4 of 5

The circus manager wants to track the address of all the clowns in the circus. Our programmer has come up with two classes – Address and Clown. Help him to bring in "has-a" relationship between the two classes.

Choose the appropriate statement from the options given, consider clown's address to be private.

- ○ self.clown_address=clown_address
- ◉ self.__clown_address=clown_address ✓
- ○ __clown_address=clown_address
- ○ clown_address=clown_address

Q5 of 5

What is the output of the following code snippet?

```
class Mobile:
    def __init__(self,brand,case):
        self.brand=brand
        self.case=case
    def display(self):
        print(self.case.color)

class Case:
    def __init__(self,color):
        self.color=color
c1=Case("Black")
c2=Case("White")
m1=Mobile("Hony",c1)
c1.color="Green"
m1.display()
```

- ○ Black
- ○ White
- ◉ Green ✓