الاسم / اسلام السيد رمزى الغرباوى

سيكشن / 1

CHAPTER 10

Multiple Choice Questions (MCQs)

1. _enter_()
2. yield
3. Maintain state and notify observers
4. Singleton
5. Stronger coupling between classes

True / False Questions

1. **True**
2. **False**
3. **True**
4. **False**
5. **True**

#Problem 1

```
import time

class Timer:

    def_enter_(self):

        self.start = time.time()
        return self


def_exit_(self, exc_type, exc_value, traceback):

end = time.time()

print(f"Execution took {end - self.start:.2f} seconds")


 with Timer():

     for i in range(1000000):

         pass
```

#Problem 2

```
def even_numbers(n):

for num in range(2, n + 1, 2):

     yield num
```

```
for num in even_numbers(10):
    print(num)
```

#Problem 3

```
def filter_positive():
    while True:
        number = yield
        if number > 0:
            print(f"Positive number: {number}")
co = filter_positive()
next(co)
co.send(-3)
co.send(5)
co.send(0
```

#Problem 4

```
class Circle:
   def draw(self):
        print("Drawing a Circle")
class Square:
    def draw(self):
        print("Drawing a Square")
def shape_factory(shape_type):
   shape_type = shape_type.lower()
   If shape_type == "circle":
      return Circle()
elif shape_type == "square":
     return Square()
else:
     raise ValueError("Unknown shape")
shape = shape_factory("circle")
shape.draw()
```

#Problem 5

```
class Observer:
   def update(self, message):
       print(f"Received update: {message}")
class Subject:
def_init_(self):
    self.observers = []


def attach(self, observer):
    self.observers. append (observer)


def detach(self, observer):
    self.observers.remove(observer)
```

```
 def notify(self, message):

     for observer in self.observers:

         observer.update.(message)

subject = Subject()

obs1, obs2 = Observer(), Observer()
subject.attach(obs1)

subject.attach(obs2)

subject.notify("Update available!")
```