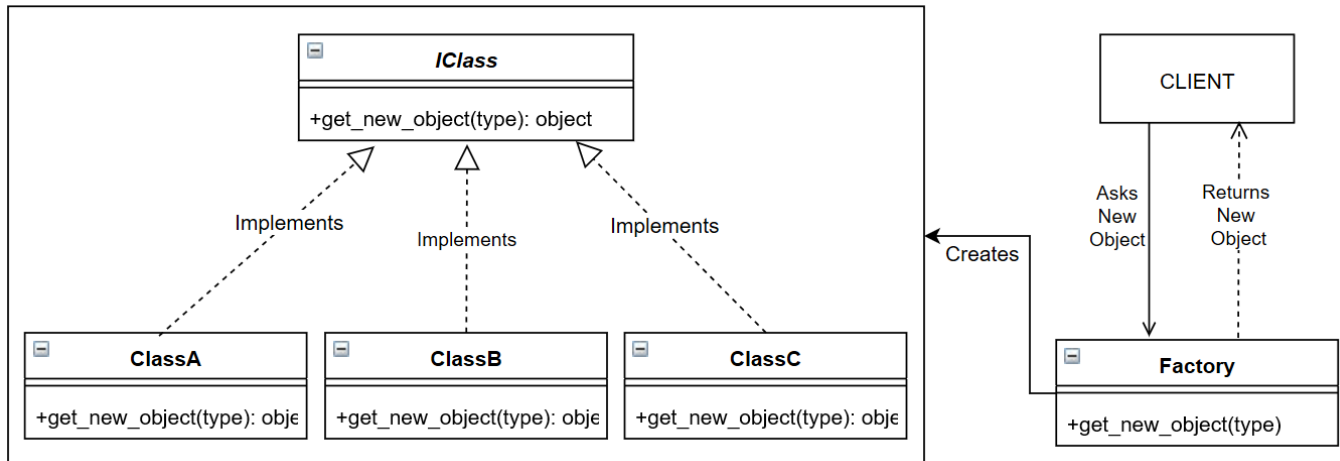


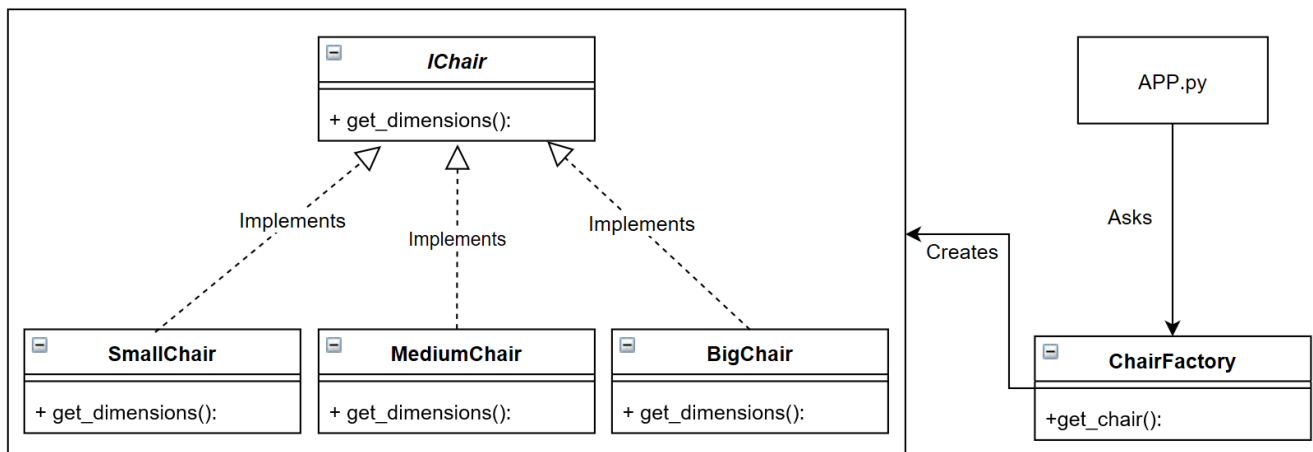
## Factory Design Pattern

The Factory Pattern is a creational pattern that defines an Interface for creating an object and defers instantiation until runtime.

Used when you don't know how many or what type of objects will be needed until or during runtime



The Factory Pattern in the context of a Chair Factory



```
class ObjectFactory:
    """Tha Factory Class"""

    @staticmethod
    def get_concrete_object(object_type):
        """A static method to get a concrete object of type class"""
        try:
            if object_type == "ObjectA":
                return ObjectA()
            if object_type == "ObjectB":
                return ObjectB()
            if object_type == "ObjectC":
                return ObjectC()
            raise AssertionError("Object Not Found")
```

# Design Patterns in Python

```
except AssertionError as _e:  
    print(_e)  
    return None
```

Each Object implements a Common Interface

```
class ObjectA(IObjectType):  
    """The Object Concrete Class which implements the IObjectType interface"""  
  
    ...
```

Request from the factory at run time

```
if __name__ == "__main__":  
    MY_OBJECT = ObjectFactory().get_concrete_object("ObjectB")  
    print(MY_OBJECT.dimensions())
```