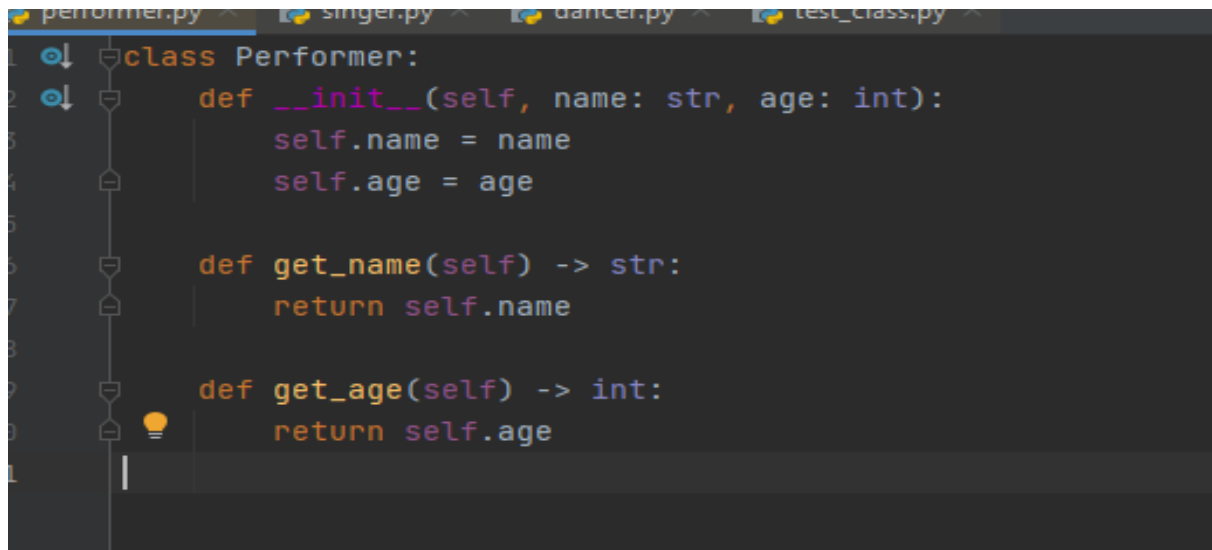


Finals Task 2. Inheritance

Mangalino, Chriselle C.
BSCS C203

Problem School Performance

performer.py(base class)

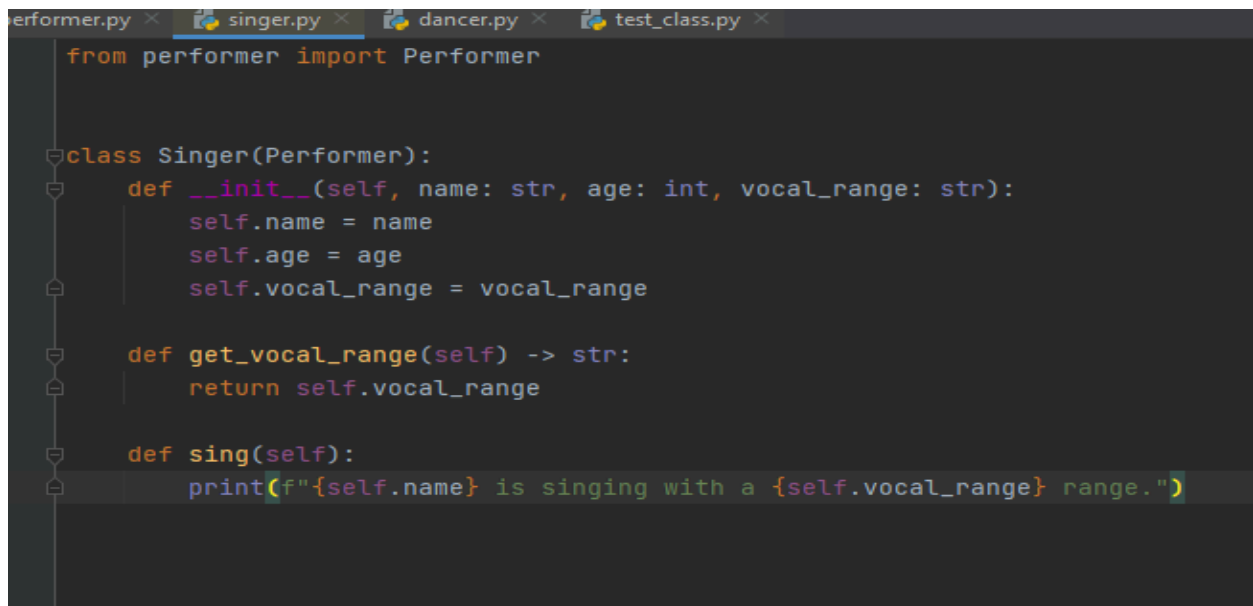
A screenshot of a code editor with a dark theme. The file explorer on the left shows four files: performer.py, singer.py, dancer.py, and test_class.py. The main editor window displays the code for performer.py. The code defines a class 'Performer' with an '.__init__' method that takes 'name' and 'age' as arguments and assigns them to 'self.name' and 'self.age'. It also has two methods: 'get_name' which returns 'self.name', and 'get_age' which returns 'self.age'.

```
class Performer:
    def __init__(self, name: str, age: int):
        self.name = name
        self.age = age

    def get_name(self) -> str:
        return self.name

    def get_age(self) -> int:
        return self.age
```

singer.py(sub class)

A screenshot of a code editor with a dark theme. The file explorer on the left shows four files: performer.py, singer.py, dancer.py, and test_class.py. The main editor window displays the code for singer.py. The code imports 'Performer' from 'performer'. It then defines a class 'Singer' that inherits from 'Performer'. The '.__init__' method for 'Singer' takes 'name', 'age', and 'vocal_range' as arguments, and assigns them to 'self.name', 'self.age', and 'self.vocal_range'. It also has two methods: 'get_vocal_range' which returns 'self.vocal_range', and 'sing' which prints a message using f-strings: 'print(f'"{self.name}" is singing with a {self.vocal_range} range.').

```
from performer import Performer

class Singer(Performer):
    def __init__(self, name: str, age: int, vocal_range: str):
        self.name = name
        self.age = age
        self.vocal_range = vocal_range

    def get_vocal_range(self) -> str:
        return self.vocal_range

    def sing(self):
        print(f'"{self.name}" is singing with a {self.vocal_range} range.')
```

dancer.py(sub class)

```
performer.py x singer.py x dancer.py x test_class.py x
from performer import Performer

class Dancer(Performer):
    def __init__(self, name: str, age: int, dance_style: str):
        self.name = name
        self.age = age
        self.dance_style = dance_style

    def get_dance_style(self) -> str:
        return self.dance_style

    def dance(self):
        print(f"{self.name} is performing {self.dance_style} dance.")
```

test_class.py – following the required test cases

```
performer.py x singer.py x dancer.py x test_class.py x
1  from performer import Performer
2  from singer import Singer
3  from dancer import Dancer
4
5  def main():
6      p = Performer( name: "John", age: 25)
7      print(p.get_name(), p.get_age())
8
9      d = Dancer( name: "Emily", age: 28, dance_style: "Ballet")
10     print(d.get_name(), d.get_age(), d.get_dance_style())
11
12     d.dance()
13
14     s = Singer( name: "Linda", age: 35, vocal_range: "Soprano")
15     print(s.get_name(), s.get_age(), s.get_vocal_range())
16
17     s.sing()
18
19
20  if __name__ == "__main__":
21     main()
```

CASE TEST 1

```
performer.py × singer.py × dancer.py × test_class.py ×
1  from performer import Performer
2  from singer import Singer
3  from dancer import Dancer
4
5  def main():
6      p = Performer(name="John", age=25)
7      print(p.get_name(), p.get_age())
```

CASE TEST 2

```
8
9      d = Dancer(name="Emily", age=28, dance_style="Ballet")
10     print(d.get_name(), d.get_age(), d.get_dance_style())
11
```

CASE TEST 3

```
11
12     d.dance()
13
```

CASE TEST 4 - IT JUST CHECK

CASE TEST 5

```
13
14     s = Singer(name="Linda", age=35, vocal_range="Soprano")
15     print(s.get_name(), s.get_age(), s.get_vocal_range())
16
```

CASE TEST 6

```
16
17     s.sing()
18
19
```

Sample Output

```
C:\Users\COMLAB\PycharmProjects\pythonProject3\venv\Scripts\python.exe  
John 25  
Emily 28 Ballet  
Emily is performing Ballet dance.  
Linda 35 Soprano  
Linda is singing with a Soprano range.  
  
Process finished with exit code 0  
|
```