Ngo, Charles Edrick P. **7OOP**

BSCS – C204 **Task 3**

Main.py

```python
from sparrow import Sparrow
from parrot import Parrot
from birdCage import BirdCage

def main():
    print("=== Individual Bird Sounds ===")
    sparrow = Sparrow()
    parrot = Parrot()

    print("Sparrow sound:")
    sparrow.make_sound()

    print("Parrot sound:")
    parrot.make_sound()

    print("\n=== Bird Cage Demo ===")
    cage = BirdCage()
    birds = [sparrow, parrot]

    print("BirdCage sounds:")
    cage.make_bird_sounds(birds)

if __name__ == "__main__":
    main()
```

Class – Bird:

```python
from abc import ABC, abstractmethod

class Bird(ABC):

    @abstractmethod
    def make_sound(self) -> None:
        pass
```

Class – Sparrow (extends Bird):

```python
from abc import ABC, abstractmethod

class Bird(ABC):

    @abstractmethod
    def make_sound(self) -> None:
        pass
```

Class – Parrot: (extends Bird):

```python
from bird import Bird

class Parrot(Bird):
    def make_sound(self) -> None:
        print("Tweet Tweet")
```

Class – BirdCage:

```python
from typing import List
from bird import Bird

class BirdCage:
    def make_bird_sounds(self, birds: List[Bird]) -> None:
        for b in birds:
            b.make_sound()
```

Sample Output: