Zabala, Rhaldynyl Brian F.
C203

Bird.py

```
from abc import ABC, abstractmethod

class Bird(ABC):

    @abstractmethod
    def make_sound(self) -> str:
        pass
```

Sparrow.py

```
from Bird import Bird

class Sparrow(Bird):

    def make_sound(self) -> str:
        return "Chirp Chirp"
```

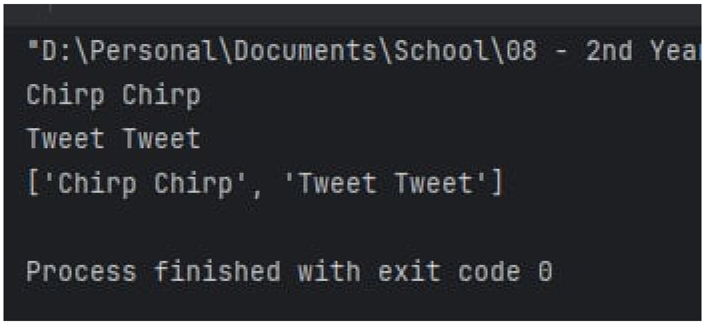
Parrot.py

```
from Bird import Bird

class Parrot(Bird):

    def make_sound(self) -> str:
        return "Tweet Tweet"
```

BirdCage.py

```
from Bird import Bird

class BirdCage:

    def make_bird_sounds(self, birds: list) -> None:
        sounds = []
        for bird in birds:
            sounds.append(bird.make_sound())
        return sounds

main.py
from Bird import Bird
from Sparrow import Sparrow
from Parrot import Parrot
from BirdCage import BirdCage

if _name_ == "_main_":
    S = Sparrow()
    P = Parrot()
    cage = BirdCage()

    print(S.make_sound())
    print(P.make_sound())
    print(cage.make_bird_sounds([S,P]))
```

Output:



```
"D:\Personal\Documents\School\08 - 2nd Yea
Chirp Chirp
Tweet Tweet
['Chirp Chirp', 'Tweet Tweet']

Process finished with exit code 0
```