

7 OOP Finals Task 3.*Simple Polymorphism Problem. Chirp and Tweet***# Class - Bird:**

```
from abc import ABC, abstractmethod
```

```
class Bird(ABC):
```

```
    @abstractmethod
    def make_sound(self):
        pass
```

Class - Sparrow (extends Bird):

```
from bird import Bird
```

```
class Sparrow(Bird):
```

```
    def make_sound(self):
        return "Chirp Chirp"
```

Class - Parrot (extends Bird):

```
from bird import Bird
```

```
class Parrot(Bird):
```

```
    def make_sound(self):
        return "Tweet Tweet"
```

Class - Bird Cage:

```
class BirdCage:
```

```
    def make_bird_sounds(self, birds):
        sounds = []
        for bird in birds:
            sounds.append(bird.make_sound())
        return sounds
```

Main:

```
from sparrow import Sparrow
```

```
from parrot import Parrot
from birdCage import BirdCage

sparrow = Sparrow()
parrot = Parrot()

print("Test Case 1")
print(sparrow.make_sound())

print(" ")
print("Test Case 2")
print(parrot.make_sound())

print(" ")
print("Test Case 3")
sparrow = Sparrow()
parrot = Parrot()
print(sparrow.make_sound())
print(parrot.make_sound())

print(" ")
print("Test Case 5")
cage = BirdCage()
print(cage.make_bird_sounds([sparrow, parrot]))
```

Sample Output:

```
Test Case 1
Chirp Chirp

Test Case 2
Tweet Tweet

Test Case 3
Chirp Chirp
Tweet Tweet

Test Case 5
['Chirp Chirp', 'Tweet Tweet']

[Program finished]
```