

Hannizel,Dumangas

BSCS – C204

Simple Polymorphism:

Source Code:

```
main.py x Bird.py Sparrow.py Parrot.py BirdCage.py
1 from Sparrow import Sparrow
2 from Parrot import Parrot
3 from BirdCage import BirdCage
4
5 # Create objects
6 sparrow = Sparrow()
7 parrot = Parrot()
8
9 # Put birds in a list
10 birds = [sparrow, parrot]
11
12 # Bird cage makes the birds produce sounds
13 cage = BirdCage()
14 cage.make_bird_sounds(birds)
```

```
main.py Bird.py x Sparrow.py Parrot.py
class Bird: 6 usages
    def make_sound(self) -> None: 1 usage
        # Abstract method
        # No specific implementation in the base class
        pass
```

```
main.py Bird.py Sparrow.py x Parrot.py BirdCage.py
1 from Bird import Bird
2
3 class Sparrow(Bird): 2 usages
4     def make_sound(self) -> None:
5         print("Chirp Chirp")
```

```
main.py Bird.py Sparrow.py Parrot.py x
1 from Bird import Bird
2
3 class Parrot(Bird): 2 usages
4     def make_sound(self) -> None:
5         print("Tweet Tweet")
```

```
main.py Bird.py Sparrow.py Parrot.py BirdCage.py x
1 from typing import List
2 from Bird import Bird
3
4 class BirdCage: 2 usages
5     def make_bird_sounds(self, birds: List[Bird]) -> None: 1 usage
6         for bird in birds:
7             bird.make_sound()
```

Sample output:

```
C:\Users\CCA_College
Chirp Chirp
Tweet Tweet
```