

### Task 1:

class Dog:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def bark(self):

print(f"{self.name} says woof!")

def sleep(self):

print(f"{self.name} is sleeping.")

dog = Dog("Buddy", 3)

dog.bark()

dog.sleep()

### Task 2:

class Dog:

def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

def bark(self):

print(f"{self.name} says woof!")

def sleep(self):

print(f"{self.name} is sleeping.")

dog = Dog("Buddy", 3)

dog.bark()

dog.sleep()

### Task 3:

class Dog:

def \_\_init\_\_(self, name, age):

```
self.name = raw_input("Enter the dog's name:")  
self.age = raw_input("Enter the dog's age:")
```

```
def bark(self):
```

```
    print(f"{self.name} is says woof!")
```

```
def sleep(self):
```

```
    print(f"{self.name} is sleeping.")
```

```
class Puppy(Dog):
```

```
    def play(self):
```

```
        print(f"{self.name} is playing with a ball!")
```

```
Puppy1 = Puppy()
```

```
Puppy1.bark()
```

```
Puppy1.sleep()
```

```
Puppy1.play()
```

Task 4:

```
class Dog:
```

```
    def __init__(self, name):
```

```
        self.name = name
```

```
    def bark(self):
```

```
        print(f"{self.name} is say woof!")
```

```
    def sleep(self):
```

```
        print(f"{self.name} is sleeping")
```

```
class Cat:
```

```
    def __init__(self, name):
```

```
        self.name = name
```

```
    def meow(self):
```

```
        print(f"{self.name} is say meow!")
```

```
    def purr(self):
```

print (Hybrid.name) is printing)

```
class Hybrid(Dog, cat):
```

```
    def __init__(self, name):
```

```
        Dog.__init__(self, name)
```

```
hybrid1 = Hybrid()
```

```
hybrid1.bark()
```

```
hybrid1.sleep()
```

```
hybrid1.meow()
```

```
hybrid1.purr()
```

Task 5:

```
class Robot:
```

```
    def __init__(self, model, energy-level):
```

```
        self.__model = model
```

```
        self.__model energy-level = energy-level
```

```
    def display_info(self):
```

```
        print(f"model: {self.__model}")
```

```
        print(f"energy-level: {self.__energy-level}")
```

```
robot1 = Robot("XR-300", 80)
```

```
robot1.display_info()
```