

Lab 5 Sample Test Case

Question 1

```
public static void testCat() {
    Cat c = new Cat("Meow", 2.0);

    System.out.println(c.getName() == "Meow");
    System.out.println(c.getWeight() == 2.0);
    c.feed();
    // The name is still the same but the weight increased by 1.0:
    System.out.println(c.getName() == "Meow");
    System.out.println(c.getWeight() == 3.0);
}
```

Question 2

```
public static void testDog() {
    Dog d = new Dog("Woof", 2.0);

    System.out.println(d.getName() == "Woof");
    System.out.println(d.getWeight() == 2.0);
    d.feed();
    // The name is still the same but the weight increased by 2.0:
    System.out.println(d.getName() == "Woof");
    System.out.println(d.getWeight() == 4.0);
}
```

Question 3

```
public static void testStudent() {
    Cat c = new Cat("Meow", 2.0);
    Student s = new Student("Philippe", c);

    System.out.println(s.getName() == "Philippe");
    System.out.println(s.getPet() == c);
}

public class Start {
    public static void main(String[] args) {
        // Unit tests
        Cat.testCat();
        Dog.testDog();
        Student.testStudent();

        // System tests, to test both classes together.
        Cat c = new Cat("Meow", 2.0);
        Student s = new Student("Philippe", c);

        System.out.println(s.getPet().getName() == "Meow");
        System.out.println(s.getPet().getWeight() == 2.0);
        // Feeding the student's pet cat:
        s.getPet().feed();
        System.out.println(s.getPet().getName() == "Meow");
        System.out.println(s.getPet().getWeight() == 3.0);
    }
}
```

Question 4

```
public static void testAnimal() {
    Animal a = new Animal("Blob", 2.0);

    System.out.println(a.getName() == "Blob");
    System.out.println(a.getWeight() == 2.0);
    a.setWeight(3.0);
    System.out.println(a.getName() == "Blob");
    System.out.println(a.getWeight() == 3.0);
}

public static void testCat() {
    Cat c = new Cat("Meow", 2.0);

    // The getName and getWeight methods are inherited from Animal.
    System.out.println(c.getName() == "Meow");
    System.out.println(c.getWeight() == 2.0);
    c.feed();
    // The name is still the same but the weight increased by 1.0:
    System.out.println(c.getName() == "Meow");
    System.out.println(c.getWeight() == 3.0);
    // The setWeight method is inherited too.
    c.setWeight(2.0);
    System.out.println(c.getName() == "Meow");
    System.out.println(c.getWeight() == 2.0);
}

public static void testDog() {
    Dog d = new Dog("Woof", 2.0);

    // The getName and getWeight methods are inherited from Animal.
    System.out.println(d.getName() == "Woof");
    System.out.println(d.getWeight() == 2.0);
    d.feed();
    // The name is still the same but the weight increased by 2.0:
    System.out.println(d.getName() == "Woof");
    System.out.println(d.getWeight() == 4.0);
    // The setWeight method is inherited too.
    d.setWeight(2.0);
    System.out.println(d.getName() == "Woof");
    System.out.println(d.getWeight() == 2.0);
}

public static void testStudent() {
    Cat c = new Cat("Meow", 2.0);
    Student s0 = new Student("Philippe", c);
    System.out.println(s0.getName() == "Philippe");
    System.out.println(s0.getPet() == c);

    // Creating a new student with a dog as pet:
    Dog d = new Dog("Woof", 2.0);
    Student s1 = new Student("Mr. Li", d);
    System.out.println(s1.getName() == "Mr. Li");
    System.out.println(s1.getPet() == d);

    // We can now also use an Animal object as a pet:
    Animal a = new Animal("Blob", 5.0);
    Student s2 = new Student("Ms. Chen", a);
    System.out.println(s2.getName() == "Ms. Chen");
    System.out.println(s2.getPet() == a);
}
```

```

public class Start {
    public static void main(String[] args) {
        // Unit tests
        // !!! Codes here written by your self !!!

        // Partial codes of System tests, to test both classes together.
        Cat c = new Cat("Meow", 2.0);
        Student s0 = new Student("Philippe", c);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 2.0);
        s0.getPet().setWeight(3.0);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 3.0);

        // Creating a new student with a dog as pet:
        Dog d = new Dog("Woof", 2.0);
        Student s1 = new Student("Mr. Li", d);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 2.0);
        s1.getPet().setWeight(3.0);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 3.0);

        // We can now also use an Animal object as a pet:
        Animal a = new Animal("Blob", 5.0);
        Student s2 = new Student("Ms. Chen", a);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 5.0);
        s2.getPet().setWeight(3.0);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 3.0);
    }
}

```

Question 5

```

public static void testBird() {
    Bird b = new Bird("Twitter", 0.5, 200.5);

    // The methods getName and getWeight are inherited from Animal.
    // The method getAltitude comes from the Bird class itself.
    System.out.println(b.getName() == "Twitter");
    System.out.println(b.getWeight() == 0.5);
    System.out.println(b.getAltitude() == 200.5);
}

public static void testStudent() {
    Cat c = new Cat("Meow", 2.0);
    Student s0 = new Student("Philippe", c);
    System.out.println(s0.getName() == "Philippe");
    System.out.println(s0.getPet() == c);

    // Creating a new student with a dog as pet:
    Dog d = new Dog("Woof", 2.0);
    Student s1 = new Student("Mr. Li", d);
    System.out.println(s1.getName() == "Mr. Li");
    System.out.println(s1.getPet() == d);

    // We can now also use an Animal object as a pet:
    Animal a = new Animal("Blob", 5.0);
    Student s2 = new Student("Ms. Chen", a);
}

```

```

System.out.println(s2.getName() == "Ms. Chen");
System.out.println(s2.getPet() == a);

// We can now also use a Bird as a pet:
Bird b = new Bird("Twitter", 0.5, 200.5);
Student s3 = new Student("Mr. Wang", b);
System.out.println(s3.getName() == "Mr. Wang");
System.out.println(s3.getPet() == b);
}

public class Start {
    public static void main(String[] args) {
        // Unit tests
        // !!! Codes here written by your self !!!

        // Partial codes of System tests, to test both classes together.
        // Creating a new student with a cat as pet:
        Cat c = new Cat("Meow", 2.0);
        Student s0 = new Student("Philippe", c);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 2.0);
        s0.getPet().setWeight(3.0);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 3.0);

        // Creating a new student with a dog as pet:
        Dog d = new Dog("Woof", 2.0);
        Student s1 = new Student("Mr. Li", d);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 2.0);
        s1.getPet().setWeight(3.0);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 3.0);

        // We can now also use an Animal object as a pet:
        Animal a = new Animal("Blob", 5.0);
        Student s2 = new Student("Ms. Chen", a);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 5.0);
        s2.getPet().setWeight(3.0);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 3.0);
        Bird b = new Bird("Twitter", 0.5, 200.5);
        Student s3 = new Student("Mr. Wang", b);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 0.5);
        s3.getPet().setWeight(3.0);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 3.0);
    }
}

```

Question 6

```

public static void testChicken() {
    Chicken c = new Chicken("Cotcot");

    // The methods getName, getWeight, and setWeight are inherited from Animal.
    // The method getAltitude is inherited from Bird.
    System.out.println(c.getName() == "Cotcot");
    System.out.println(c.getWeight() == 5.0);
}

```

```

        System.out.println(c.getAltitude() == 0.0);
        c.setWeight(2.0);
        System.out.println(c.getName() == "Cotcot");
        System.out.println(c.getWeight() == 2.0);
        System.out.println(c.getAltitude() == 0.0);
    }

    public static void testStudent() {
        Cat c = new Cat("Meow", 2.0);
        Student s0 = new Student("Philippe", c);
        System.out.println(s0.getName() == "Philippe");
        System.out.println(s0.getPet() == c);

        // Creating a new student with a dog as pet:
        Dog d = new Dog("Woof", 2.0);
        Student s1 = new Student("Mr. Li", d);
        System.out.println(s1.getName() == "Mr. Li");
        System.out.println(s1.getPet() == d);

        // We can now also use an Animal object as a pet:
        Animal a = new Animal("Blob", 5.0);
        Student s2 = new Student("Ms. Chen", a);
        System.out.println(s2.getName() == "Ms. Chen");
        System.out.println(s2.getPet() == a);

        // We can now also use a Bird as a pet:
        Bird b = new Bird("Twitter", 0.5, 200.5);
        Student s3 = new Student("Mr. Wang", b);
        System.out.println(s3.getName() == "Mr. Wang");
        System.out.println(s3.getPet() == b);

        // We can now also use a Chicken as a pet:
        Chicken ch = new Chicken("Cotcot");
        Student s4 = new Student("Ms. Liu", ch);
        System.out.println(s4.getName() == "Ms. Liu");
        System.out.println(s4.getPet() == ch);
    }

    public class Start {
        public static void main(String[] args) {
            // Unit tests
            // !!! Codes here written by your self !!!

            // Partial codes of System tests, to test both classes together.
            // Creating a new student with a cat as pet:
            Cat c = new Cat("Meow", 2.0);
            Student s0 = new Student("Philippe", c);
            System.out.println(s0.getPet().getName() == "Meow");
            System.out.println(s0.getPet().getWeight() == 2.0);
            s0.getPet().setWeight(3.0);
            System.out.println(s0.getPet().getName() == "Meow");
            System.out.println(s0.getPet().getWeight() == 3.0);

            // Creating a new student with a dog as pet:
            Dog d = new Dog("Woof", 2.0);
            Student s1 = new Student("Mr. Li", d);
            System.out.println(s1.getPet().getName() == "Woof");
            System.out.println(s1.getPet().getWeight() == 2.0);
            s1.getPet().setWeight(3.0);
            System.out.println(s1.getPet().getName() == "Woof");
            System.out.println(s1.getPet().getWeight() == 3.0);
        }
    }

```

```

        // We can now also use an Animal object as a pet:
        Animal a = new Animal("Blob", 5.0);
        Student s2 = new Student("Ms. Chen", a);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 5.0);
        s2.getPet().setWeight(3.0);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 3.0);
        Bird b = new Bird("Twitter", 0.5, 200.5);
        Student s3 = new Student("Mr. Wang", b);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 0.5);
        s3.getPet().setWeight(3.0);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 3.0);

        // We can now also use a Chicken as a pet:
        Chicken ch = new Chicken("Cotcot");
        Student s4 = new Student("Ms. Liu", ch);
        System.out.println(s4.getPet().getName() == "Cotcot");
        System.out.println(s4.getPet().getWeight() == 5.0);
        s4.getPet().setWeight(3.0);
        System.out.println(s4.getPet().getName() == "Cotcot");
        System.out.println(s4.getPet().getWeight() == 3.0);
    }
}

```

Question 7

```

public static void testLivingThing() {
    LivingThing l = new LivingThing("Glop");
    System.out.println(l.getName() == "Glop");
}

public static void testAnimal() {
    Animal a = new Animal("Blob", 2.0);

    // The method getName is inherited from LivingThing.
    // The methods getWeight and setWeight come from Animal itself.
    System.out.println(a.getName() == "Blob");
    System.out.println(a.getWeight() == 2.0);
    a.setWeight(3.0);
    System.out.println(a.getName() == "Blob");
    System.out.println(a.getWeight() == 3.0);
}

public static void testStudent() {
    Cat c = new Cat("Meow", 2.0);
    Student s0 = new Student("Philippe", c);
    // The method getName is inherited from LivingThing.
    // The method getPet is from the Student class itself.
    System.out.println(s0.getName() == "Philippe");
    System.out.println(s0.getPet() == c);

    Dog d = new Dog("Woof", 2.0);
    Student s1 = new Student("Mr. Li", d);
    System.out.println(s1.getName() == "Mr. Li");
    System.out.println(s1.getPet() == d);

    Animal a = new Animal("Blob", 5.0);
    Student s2 = new Student("Ms. Chen", a);
}

```

```

System.out.println(s2.getName() == "Ms. Chen");
System.out.println(s2.getPet() == a);

Bird b = new Bird("Twitter", 0.5, 200.5);
Student s3 = new Student("Mr. Wang", b);
System.out.println(s3.getName() == "Mr. Wang");
System.out.println(s3.getPet() == b);

Chicken ch = new Chicken("Cotcot");
Student s4 = new Student("Ms. Liu", ch);
System.out.println(s4.getName() == "Ms. Liu");
System.out.println(s4.getPet() == ch);
}

public class Start {
    public static void main(String[] args) {
        // Unit tests
        // !!! Codes here written by your self !!!

        // Partial codes of System tests, to test both classes together.
        // Creating a new student with a cat as pet:
        Cat c = new Cat("Meow", 2.0);
        Student s0 = new Student("Philippe", c);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 2.0);
        s0.getPet().setWeight(3.0);
        System.out.println(s0.getPet().getName() == "Meow");
        System.out.println(s0.getPet().getWeight() == 3.0);

        // Creating a new student with a dog as pet:
        Dog d = new Dog("Woof", 2.0);
        Student s1 = new Student("Mr. Li", d);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 2.0);
        s1.getPet().setWeight(3.0);
        System.out.println(s1.getPet().getName() == "Woof");
        System.out.println(s1.getPet().getWeight() == 3.0);

        // We can now also use an Animal object as a pet:
        Animal a = new Animal("Blob", 5.0);
        Student s2 = new Student("Ms. Chen", a);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 5.0);
        s2.getPet().setWeight(3.0);
        System.out.println(s2.getPet().getName() == "Blob");
        System.out.println(s2.getPet().getWeight() == 3.0);
        Bird b = new Bird("Twitter", 0.5, 200.5);
        Student s3 = new Student("Mr. Wang", b);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 0.5);
        s3.getPet().setWeight(3.0);
        System.out.println(s3.getPet().getName() == "Twitter");
        System.out.println(s3.getPet().getWeight() == 3.0);

        // We can now also use a Chicken as a pet:
        Chicken ch = new Chicken("Cotcot");
        Student s4 = new Student("Ms. Liu", ch);
        System.out.println(s4.getPet().getName() == "Cotcot");
        System.out.println(s4.getPet().getWeight() == 5.0);
        s4.getPet().setWeight(3.0);
        System.out.println(s4.getPet().getName() == "Cotcot");
        System.out.println(s4.getPet().getWeight() == 3.0);
    }
}

```

} }