Quiz of OOP

Question #1

You're learning about encapsulation in Java. Design a simple Java program for managing a student's information. Create a **Student** class with private fields for the student's name, age, and grade. Ensure that these fields cannot be directly accessed from outside the class. Provide methods to set and get these details. Additionally, include a method to display the student's information.

Your program should demonstrate encapsulation by hiding the internal state of the **Student** class and providing controlled access to its fields.

Question # 2

1)

class Animal {

void makeSound() {

System.***out***.println("Some sound");

}

}

class Dog extends Animal {

void makeSound() {

System.***out***.println("Woof");

}

}

class Cat extends Animal {

void makeSound() {

System.***out***.println("Meow");

}

}

public class Main {

public static void main(String[] args) {

Animal a1 = new Dog();

Animal a2 = new Cat();

Dog d1 = (Dog) a1;

Cat c1 = (Cat) a2;

a1.makeSound();

a2.makeSound();

d1.makeSound();

c1.makeSound();

}

}

2)

class Person {

private String name;

private int age;

public Person(String name, int age) {

this.name = name;

this.age = age;

}

public void display() {

System.***out***.println("Name: " + name);

System.***out***.println("Age: " + age);

}

public void setName(String name) {

if (name != null && !name.isEmpty()) {

this.name = name;

} else {

System.***out***.println("Invalid name.");

}

}

public void setAge(int age) {

if (age >= 0 && age <= 120) {

this.age = age;

} else {

System.***out***.println("Invalid age.");

}

}

}

public class Main {

public static void main(String[] args) {

Person person = new Person("John", 30);

person.display();

person.setName("Alice");

person.setAge(150);

person.display();

}

}