

Zid'Avwa Al Bari'i

244107020083 / TI – 2I / 26

Jobsheet 10

Exercise 1

Codes

```
src > J Animal.java > Animal
1 public abstract class Animal {
2     private int age; // The value of the field Animal.age is not used
3
4     protected Animal() { this.age = 0; }
5     public void ageUp() { this.age++; }
6     public abstract void moving();
7 }
```

```
src > J Cat.java > Cat > moving()
1 public class Cat extends Animal {
2     public void moving() {
3         System.out.println("Walk using four legs, \nTAP..TAP..\n");
4     }
5 }
```

```
src > J Fish.java > Fish
1 public class Fish extends Animal {
2     public void moving() {
3         System.out.println("Swim using fins, \nwash..wash..\n");
4     }
5 }
```

```
src > J People.java > ...
1 public class People {
2     private String name;
3     private Animal pet;
4
5     public People(String name) { this.name = name; }
6     public void setPet(Animal pet) { this.pet = pet; }
7
8     public void walkThePet() {
9         System.out.println("My Name is " + this.name);
10        System.out.println("My Pet walks by: ");
11        this.pet.moving();
12        System.out.println("-----");
13    }
14 }
```

```
src > J Main.java > Main
1 public class Main {
2     Run | Debug
3     public static void main(String[] args) {
4         Cat wildCat = new Cat();
5         Fish dolphin = new Fish();
6
7         People ani = new People(name:"Ani");
8         People budi = new People(name:"Budi");
9
10        ani.setPet(wildCat);
11        budi.setPet(dolphin);
12
13        ani.walkThePet();
14        budi.walkThePet();
15    }
16 }
```

Questions & Answers:

Discussion: Is it permissible if a class that extends an abstract class does not implement the abstract method in its parent class? Prove!

No, it's not permissible, unless the subclass is also declared abstract. If a subclass doesn't implement the abstract method, Java forces it to become abstract too.

```

src> J Chicken.java > Chicken
1 public class Chicken extends Animal {
2     public void eating() { System.out.println("Chicken is eating"); }
3 }
12 ani.setPet(chicken);
13
14 ani.walkThePet();
15 budi.walkThePet();
16 ani.feed();
17
18 public void feed() {
19     System.out.println("My Name is " + this.name);
20     System.out.println("My Pet eating a: ");
21     this.chicken.eating();
22     System.out.println("-----");
23 }

```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The type Chicken must implement the inherited abstract method Animal.moving()
at Chicken.moving(Chicken.java:1)

1. Provide a related explanation of the program above

The program demonstrates inheritance and polymorphism.

Animal defines an abstract method moving().

Cat and Fish implement this method differently.

People holds an Animal reference and calls pet.moving() — showing dynamic behavior depending on the object's actual type (Cat or Fish).

2. Show the compilation results of the program and give a brief explanation if the moving method () is changed to method abstract!

If moving() in Animal is no longer abstract (given a body), the program still compiles and runs.

Subclasses can override it or use the default version.

Polymorphism still works, but if no override exists, the output will be from Animal's method.

```

6 // public abstract void moving();
7 public void moving() { System.out.println(x:"Animal is moving..."); }

```

```

My Name is Ani
My Pet walks by:
Animal is moving...
-----
My Name is Budi
My Pet walks by:
Swim using FINS, "wush..wush.."
-----

```

3. Show the results of the program compilation and provide a brief explanation if there is no overriding of the moving method()

If Cat and Fish remove their moving() methods while Animal keeps it abstract, compilation fails because abstract methods must be implemented.

To fix it, Cat and Fish would also need to be declared abstract. Will show the same error as the Chicken Class from the discussion.

4. Show the results of the program compilation and provide a brief explanation if the abstract method move() declared in the Fish Class

If Fish declares moving() as abstract, then Fish must also be abstract and cannot be instantiated.

Main would fail at new Fish() because abstract classes can't be created directly, a concrete subclass of Fish must implement moving() first.

```

Main.java src 1
Cannot instantiate the type Fish Java(16777373) [Ln 4, Col 28]

```