```java
public class Monster {
    protected String name;

    public Monster(String name){
        this.name = name;
    }

    public String attack(){
//        return "Attack with ";
        return "I don't know how to attack!";
    }

}

```

```java
public class FireMonster extends  Monster{
    public FireMonster(String name){
        super(name);
    }

    @Override
    public String attack() {
        return "Attack with fire!";
    }
}

```

```java
public class TestMonster {
    public static void main(String[] args){
        // Declare instances of the superclass, substituted by subclasses.
        Monster m1 = new FireMonster("r2u2");   // upcast
        Monster m2 = new WaterMonster("u2r2");  // upcast
        Monster m3 = new StoneMonster("r2r2");  // upcast


        // Invoke the actual implementation
        System.out.println(m1.attack());  // Run FireMonster's attack()
        System.out.println(m2.attack());  // Run WaterMonster's attack()
        System.out.println(m3.attack());  // Run StoneMonster's attack()


        // m1 dies, generates a new instance and re-assign to m1.
        m1 = new StoneMonster("a2b2");  // upcast
        System.out.println(m1.attack());  // Run StoneMonster's attack()


        // We have a problem here!!!
        Monster m4 = new Monster("u2u2");
        System.out.println(m4.attack());

    }
}

```

```java
public class StoneMonster extends Monster{
    public StoneMonster(String name){
        super(name);
    }

    @Override
    public String attack() {
        return "Attack with stones!";    }
}

```

```java
public class WaterMonster extends Monster{
    public WaterMonster(String name){
        super(name);
    }

    @Override
    public String attack() {
        return "Attack with water!";
    }
}

```