Code Smells:

1.)

Before:

```
public class MageCow extends Cow {
    public MageCow(int health, int cost, int attackSpeed, int damage, int level, float range) {
        super(health, cost, attackSpeed, damage, level, range);
    }
    public void attackEnemy() { }
}
```

This code represents an **OO Abuser: Refused Bequest** because the only motivation to create this inherited class was to super the variables passed through. No other adjustments were created for this specific cow. This code smell exists because splash damage, a unique feature for a Mage Cow, was never implemented.

After:

```
public class MageCow extends Cow {
    public MageCow(int health, int cost, int attackSpeed, int damage, int level, float range) {
        super(health, cost, attackSpeed, damage, level, range);
        int splashDamage = this.getDamage();
    }
    public void attackEnemy() { }
}
```

The code was adjusted to include the previously mentioned feature of splashDamage. This corrects the **OO Abuser: Refused Bequest** code smell as there is now purpose for using inheritance in MageCow.

2.)

Before:

```java
public abstract class Cow {
    private int health = 0;
    private int cost = 0;
    private int attackSpeed = 0;
    private int damage = 0;
    private int level = 0;
    private float range = 0;
    private int splashDamage = 0;
    private boolean canPenetrate = false;
    public Cow(int health, int cost, int attackSpeed, int damage, int level, float range) {
        this.health = health;
        this.cost = cost;
        this.attackSpeed = attackSpeed;
        this.damage = damage;
        this.level = level;
        this.range = range;
    }
    abstract void attackEnemy();
    public int getHealth() { return health; }
    public int getCost() { return cost; }
    public int getAttackSpeed() { return attackSpeed; }
    public int getDamage() { return damage; }
    public int getLevel() { return level; }
    public float getRange() { return range; }
    public int getSplashDamage() {return splashDamage;}
    public boolean getCanPenetrate() {return canPenetrate;}
}
```

This code is an example of **Dispensables: Data Class** where the existence of this abstract class Cow is to set the values for the cow objects and have getters and setters for this cow class. This class exists as a result of us needing a cow class where we can set the values for the cow variables, however as a result it is a shell class that cannot function on its own.

After:

```java
public class Cow {
    private int x, y, left, top, bottom, right;
    private int cowBodyColor = Color.BLACK;
    private int towerDamage;
    private int towerRange;
    private int ID;
    private int towerType = 0;// 0 = basic, 1 = mage, 2 = cannon, 3 = cc

    public Cow(int x, int y, int ID, int towerType) {
        this.ID = ID;
        this.towerType = towerType;
        this.x = x;
        this.y = y;
        left = x - 50;
        top = y - 50;
        bottom = y + 50;
        right = x + 50;
    }
```

In order to fix this code, we make cow a necessary class and we change the java class so that it is relevant and is not used for just getters and setters. In this case we changed Cow so that it is used to place down / draw out the Cow objects.

3.)

Before:

```
public class Shop extends AppCompatActivity {
    private int poolOfBasicCows = 50;
    private int poolOfCannonCows = 50;
    private int poolOfMageCows = 50;
    private int poolOfCCCows = 50;
    private int basicCowsOwned = 0;
    private int cannonCowsOwned = 0;
    private int mageCowsOwned = 0;
    private int ccCowsOwned = 0;
    private int totalPoolOfCows = 0;
    private int money = 0;
    private int cowPrice = 0;
    private static boolean firstTime = true;
    private static int slot1num = 0;
    private static int slot2num = 0;
    private static int slot3num = 0;
    private int drawableInt;
    private boolean canBuy = false;
    private int slotsUsed;
```

This code is an example of a **Bloater: Large class**. In this class there are many declared instance variables and the class is doing too much. This class exists with this many declared instance variables because we tried to have our shop class do more than it needed too.

After:

```
public class Shop extends AppCompatActivity {
    private int poolOfBasicCows = 50;
    private int poolOfCannonCows = 50;
    private int poolOfMageCows = 50;
    private int poolOfCCCows = 50;
    private int totalPoolOfCows = 0;
    private int money = 0;
    private int cowPrice = 0;
    private static boolean firstTime = true;
    private static int slot1num = 0;
    private static int slot2num = 0;
    private static int slot3num = 0;
    private int drawableInt;
    private boolean canBuy = false;
```

By reducing the amount of instance variables down to what is absolutely necessary in this class, we can reduce the amount of bloat in this class. Since this is our shop class, the class will be larger than most classes, but it has been optimized to not be an over bloated class now.

4.)

Before:

```
public class CannonCow extends Cow {
    public CannonCow(int health, int cost, int attackSpeed, int damage, int level, float range) {
        super(health, cost, attackSpeed, damage, level, range);
    }
    public void attackEnemy() { }
}
```

CannonCow is an example of a **lazy class**. The only thing that this class does is inherit the values from the cow class and then super those values. This class exists because cannonCow is meant to be able to penetrate camouflaged farmers, however, the cannonCow does not have the ability to do this currently.

After:

```java
public class CannonCow extends Cow {
    public CannonCow(int health, int cost, int attackSpeed, int damage, int level, float range) {
        super(health, cost, attackSpeed, damage, level, range);
        boolean canPenetrate = this.getCanPenetrate();

        canPenetrate = true;
    }
    public void attackEnemy() { }
}
```

By implementing the canPenetrate feature, whether or not the cow can see/attack camouflaged/lead farmers, the cannonCow class is not an example of a lazy class. Since there is a purpose to having the cannonCow class by turning the default false canPenetrate boolean variable to true.