

# Answer Key — Week 10 Parsons Problems

For TAs only — do not distribute

---

## Problem 1 — Drum

```
abstract class Instrument {
    abstract int volume();

    void printVolume() { System.out.println("Volume: " + volume()); }
}

public class Drum extends Instrument {
    int volume() { return 8; }

    public static void main(String[] args) {
        Drum d = new Drum();
        d.printVolume();
    }
}
```

**DISTRACTOR** `int volume();`

Missing the `abstract` keyword — a method with no body is only legal in an interface, or in a class explicitly marked `abstract` with the method itself marked `abstract`.

**DISTRACTOR** `Instrument i = new Instrument();`

Tries to instantiate `Instrument` directly — abstract classes cannot be instantiated with `new`; only concrete subclasses like `Drum` can be.

## Problem 2 — Wizard

```
interface SpellCaster { int castFireball(); }
interface Healer { int castHeal(); }

public class Wizard implements SpellCaster, Healer {
    public int castFireball() { return 40; }
    public int castHeal() { return 25; }

    public static void main(String[] args) {
        Wizard w = new Wizard();
        System.out.println("Fireball damage: " + w.castFireball() + "\nHeal amount: " +
w.castHeal());
    }
}
```

**DISTRACTOR** `public class Wizard extends SpellCaster, Healer {`

Wrong keyword — `extends` is for subclassing a single class, not for interfaces. A class that implements interfaces must use `implements`, e.g. `class Wizard implements SpellCaster, Healer`.

**DISTRACTOR** `public int CastFireball() { return 40; }`

Capitalized `CastFireball` instead of `castFireball` — Java is case-sensitive, so this does not implement `SpellCaster`'s `castFireball()` method at all; it just adds an unrelated, unused method.