Interview Q&A

1. Difference between FileReader and BufferedReader?

`FileReader` reads one character at a time directly from file → slower.
`BufferedReader` wraps it with a buffer and supports `readLine()` → faster and efficient for line-based reading.

2. What is try-with-resources?

A Java 7+ feature ensuring automatic closing of resources implementing `AutoCloseable`.
Syntax:

```
try (FileWriter writer = new FileWriter("file.txt")) {
   // code
}
```

3. How to handle IOException?

Use try-catch or try-with-resources.
Always log or print stack trace to debug:

```
try { ... } catch (IOException e) { e.printStackTrace(); }
```

4. Checked vs Unchecked Exceptions

- **Checked:** Verified at compile-time (e.g., IOException)
- **Unchecked:** Runtime exceptions (e.g., NullPointerException)

5. How does file writing work in Java?

Use `FileWriter` → `write()` → `flush()` → `close()`.
Append mode (`true`) keeps existing content, overwrite replaces it.

6. Append vs Overwrite Mode

- **Append:** `new FileWriter(file, true)` → adds new data
- **Overwrite:** `new FileWriter(file)` → clears file before writing

7. What is Exception Propagation?

When a method throws an exception up the call stack using `throws`.
Allows caller to handle it instead.

8. How to log exceptions?

- `e.printStackTrace()`
- Use logging frameworks (`Log4j`, `SLF4J`)
- Include timestamp, message, and context

9. What is a Stack Trace?

A list of method calls leading to an exception — shows file, class, and line number.

10. When to use finally block?

Use `finally` for cleanup, resource release, or logs — always executes except when JVM exits.