Java Mastery Notes

1. Exception Handling in Java
- Definition: Mechanism to handle runtime errors.
- Types of Exceptions:
  - Checked Exception (e.g., IOException)
  - Unchecked Exception (e.g., ArithmeticException)
- Keywords: try, catch, finally, throw, throws
Example:
```
try {
    int result = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Cannot divide by zero");
} finally {
    System.out.println("Execution completed");
}
```

2. Generics & Wrapper Classes
a) Generics
- Enable classes, interfaces, and methods to operate on types specified by the programmer.
Example:
```
ArrayList<String> list = new ArrayList<>();
list.add("Hello");
```

b) Wrapper Classes & Autoboxing/Unboxing
- Convert primitives to objects and vice versa.
- Autoboxing: Primitive to Object
- Unboxing: Object to Primitive
Example:
```
Integer num = 10; // Autoboxing
int n = num;      // Unboxing
```

3. Collection Framework
a) ArrayList & LinkedList
- ArrayList: Dynamic array, fast random access, slow insertion/deletion in middle.
- LinkedList: Doubly-linked list, fast insertion/deletion, slower random access.
Example:
```
ArrayList<Integer> arr = new ArrayList<>();
```

LinkedList<Integer> list = new LinkedList<>();

b) Collection Interface
- Root interface in the collection hierarchy.
- Common methods: add(), remove(), size(), contains()

c) Vector & Stack
- Vector: Thread-safe ArrayList.
- Stack: LIFO structure.
Example:
Stack<Integer> stack = new Stack<>();
stack.push(1);
stack.pop();

d) Queue & Sets
- PriorityQueue: Elements sorted according to priority.
- ArrayDeque: Double-ended queue.
- HashSet: Stores unique elements.
Example:
PriorityQueue<Integer> pq = new PriorityQueue<>();
HashSet<String> set = new HashSet<>();

e) Map Interface & Comparators/Comparable
- Map: Key-Value pairs (HashMap, TreeMap).
- Comparable: Defines natural order.
- Comparator: Custom sorting logic.
Example:
```
class Student implements Comparable<Student> {
    int marks;
    public int compareTo(Student s) { return this.marks - s.marks; }
}
```

f) Lambda Expressions
- Short syntax for functional interfaces.
Example:
List<Integer> nums = Arrays.asList(1,2,3);
nums.forEach(n -> System.out.println(n));

4. File Handling in Java

- File Operations: Create, read, write files.
- Classes: File, FileReader, FileWriter, BufferedReader, BufferedWriter

Example:

```
File file = new File("example.txt");
if(file.createNewFile()) System.out.println("File created");

FileWriter writer = new FileWriter(file);
writer.write("Hello World");
writer.close();
```