Java Practical Revision Notes

Part 1: Clean & Simple Version

1. Generic Class

A Generic Class allows you to define a class that can work with different data types using .

```
class Box<T> {
    private T item;
    public void setItem(T item) { this.item = item; }
    public T getItem() { return item; }
}

public class Main {
    public static void main(String[] args) {
        Box<Integer> intBox = new Box<>();
        intBox.setItem(10);
        System.out.println("Integer value: " + intBox.getItem());

        Box<String> strBox = new Box<>();
        strBox.setItem("Hello");
        System.out.println("String value: " + strBox.getItem());
    }
}
```

2. Collections (ArrayList, LinkedList)

```
import java.util.*;

public class ArrayListExample {
    public static void main(String[] args) {
        List<String> students = new ArrayList<>();
        students.add("Ravi");
        students.add("Sita");
        students.add("Manu");
        students.add("Lina");
        students.add("Kiran");

        System.out.println("Student List:");
        for (String s : students)
            System.out.println(s);
    }
}
```

```
LinkedList<String> list = new LinkedList<>();
list.add("A");
list.add("B");
list.addFirst("Start");
System.out.println(list);
```

3. TCP & UDP Communication

```
// TCP Server
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) throws IOException {
        ServerSocket ss = new ServerSocket(5000);
        Socket s = ss.accept();
        BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
        PrintWriter out = new PrintWriter(s.getOutputStream(), true);
        String msg = in.readLine();
        System.out.println("Client: " + msg);
        out.println("Received: " + msg);
        ss.close();
    }
}

// TCP Client
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) throws IOException {
        Socket s = new Socket("localhost", 5000);
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter out = new PrintWriter(s.getOutputStream(), true);
        BufferedReader br = new BufferedReader(new InputStreamReader(s.getInputStream()));
        System.out.print("Enter message: ");
        String msg = in.readLine();
        out.println(msg);
        System.out.println("Server: " + br.readLine());
        s.close();
    }
}
```

4. Lambda & Block Lambda

```
interface Square {
    int calc(int x);
}

public class LambdaDemo {
    public static void main(String[] args) {
        Square s = (x) -> x * x;
        System.out.println("Square: " + s.calc(5));
    }
}

interface Factorial {
    int compute(int n);
}

public class BlockLambda {
    public static void main(String[] args) {
        Factorial f = (n) -> {
            int result = 1;
            for (int i = 1; i <= n; i++)
                result *= i;
            return result;
        };
        System.out.println("Factorial of 5: " + f.compute(5));
    }
}
```

Part 2: Stylish & Colorful Version

Generic Class (Colored)

```
// Example: Generic Class
class Box<T> {
    private T item;
    public void setItem(T item) { this.item = item; }   // Setter
    public T getItem() { return item; }                 // Getter
}

public class Main {
    public static void main(String[] args) {
        Box<Integer> intBox = new Box<>();   // Integer type
        intBox.setItem(10);
        System.out.println("Integer: " + intBox.getItem());

        Box<String> strBox = new Box<>();    // String type
        strBox.setItem("Hello");
        System.out.println("String: " + strBox.getItem());
    }
}
```

Collections (Colored)

```
// Using ArrayList
import java.util.*;
public class ArrayListExample {
    public static void main(String[] args) {
        List<String> list = new ArrayList<>();
        list.add("A");
        list.add("B");
        list.add("C");

        list.forEach(x -> System.out.println(x)); // Lambda in collections
    }
}
```

Lambda Expression (Colored)

```
// Simple Lambda Example
interface MathOperation {
    int operate(int a, int b);
}

public class LambdaExample {
    public static void main(String[] args) {
        MathOperation add = (a, b) -> a + b;
        System.out.println("Sum: " + add.operate(5, 3));
    }
}
```