Java Notes – From Basics to Method Overloading

1. Introduction to Java
- Java is a high-level, object-oriented programming language.
- Platform-independent: Write once, run anywhere (WORA).
- Uses JVM (Java Virtual Machine) to run programs.

2. Setting Up Java
- Install JDK (Java Development Kit).
- Set PATH and JAVA_HOME environment variables.
- Use an IDE (Eclipse, IntelliJ, VS Code) or simple text editor.

3. Java Basics
- Structure of a Java program:

```
public class Main {
public static void main(String[] args) {
System.out.println("Hello, World!");
}
}
```

- Every application must have a main method.

4. Variables & Data Types
- Variable: A container for storing data.
- Data types:
- Primitive: byte, short, int, long, float, double, char, boolean
- Non-primitive: Strings, Arrays, Objects
- Example:

```
int age = 25;
String name = "John";
```

5. Operators
- Arithmetic: +, -, *, /, %
- Relational: ==, !=, >, <, >=, <=
- Logical: &&, ||, !
- Assignment: =, +=, -=, *=, /=
- Increment/Decrement: ++, --

6. Control Statements
- if, if-else, if-else-if ladder
- switch statement

Example:

```
if(age > 18) {
System.out.println("Adult");
} else {
System.out.println("Minor");
}
```

7. Loops
- for, while, do-while

Example:

```
for(int i=0; i<5; i++) {
System.out.println(i);
}
```

8. Arrays
- Collection of elements of the same type.

Example:

```
int[] numbers = {1, 2, 3, 4, 5};
System.out.println(numbers[0]); // Output: 1
```

9. Methods
- Block of code executed when called.
Example:

```
void greet() {
System.out.println("Hello");
}
```

10. Method Overloading
- Two or more methods with the same name but different parameters.
Example:

```
void sum(int a, int b) {
System.out.println(a + b);
}
void sum(double a, double b) {
System.out.println(a + b);
}
```