Reading 18: Parameters

# Exercise 1: Summarize

Parameters are integral to passing data between functions, with their values and names varying significantly based on how they are passed, referenced, initialized, and the language rules.

# Exercise 2: Demonstrate & Explain

public class Main {

public static void main(String[] args) {

int p = 10;

Object o = new Object();

System.out.println("Main P address: " + System.identityHashCode(p));

System.out.println("Main O address: " + System.identityHashCode(o));

PassByValue(p, o);

}

public static void PassByValue(int p, Object o) {

System.out.println("P address: " + System.identityHashCode(p));

System.out.println("O address: " + System.identityHashCode(o));

}

}

// Output:

// Main P address: 31168322

// Main O address: 17225372

// P address: 31168322

// O address: 17225372

As the output shows the addresses of the variables are the same in both the main function and the called function, we can conclude that Java uses pass-by-value for its parameters. This is because, for primitives, Java passes a copy of the actual value, and for objects, it passes a copy of the reference to the object. The addresses match because the reference itself is copied, not the object. Therefore, any changes to the object within the method affect the original object, but the reference remains the same, demonstrating Java's pass-by-value nature.

# Exercise 3: Inquire

Can you explain the difference between pass-by-value and pass-by-reference, and how Java handles parameter passing?