

# Introduction to Java for C++ Programmers

## JAC444

### **Week 03**

### **Methods and Arrays**

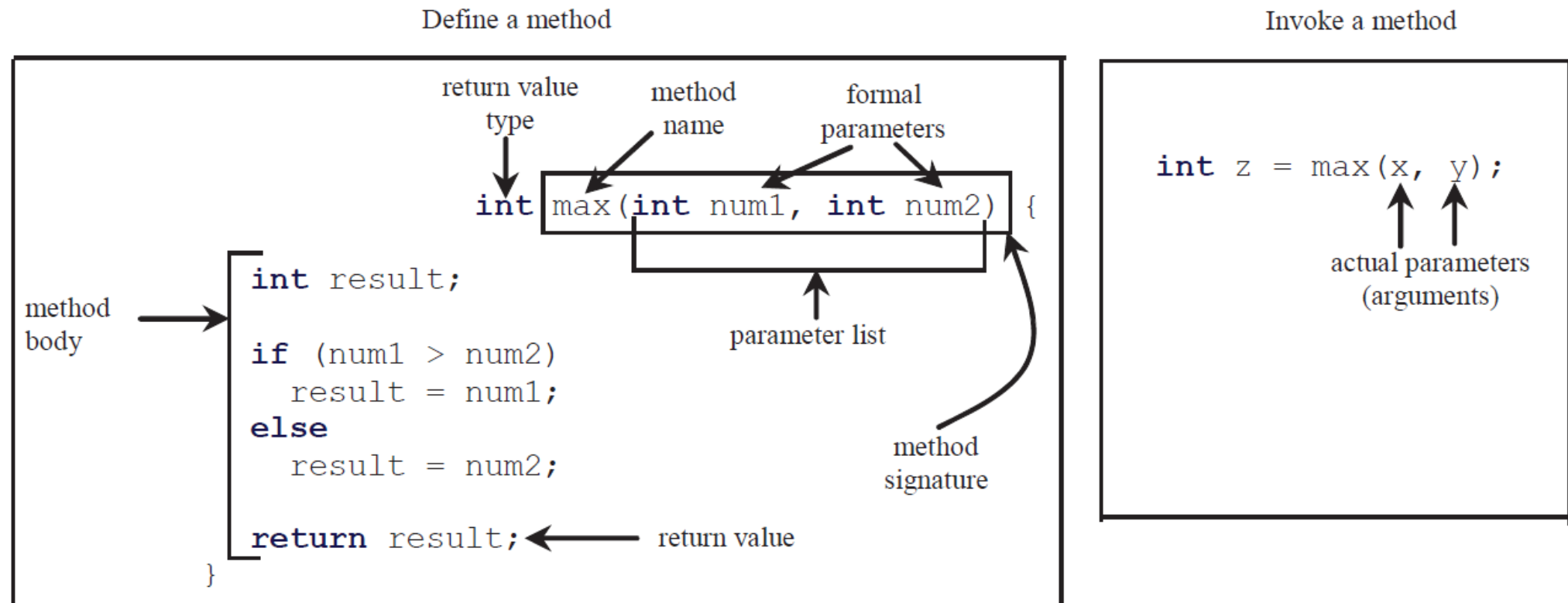
Instructor

Hossein Pourmodheji

hossein.pourmodheji@senecacollege.ca

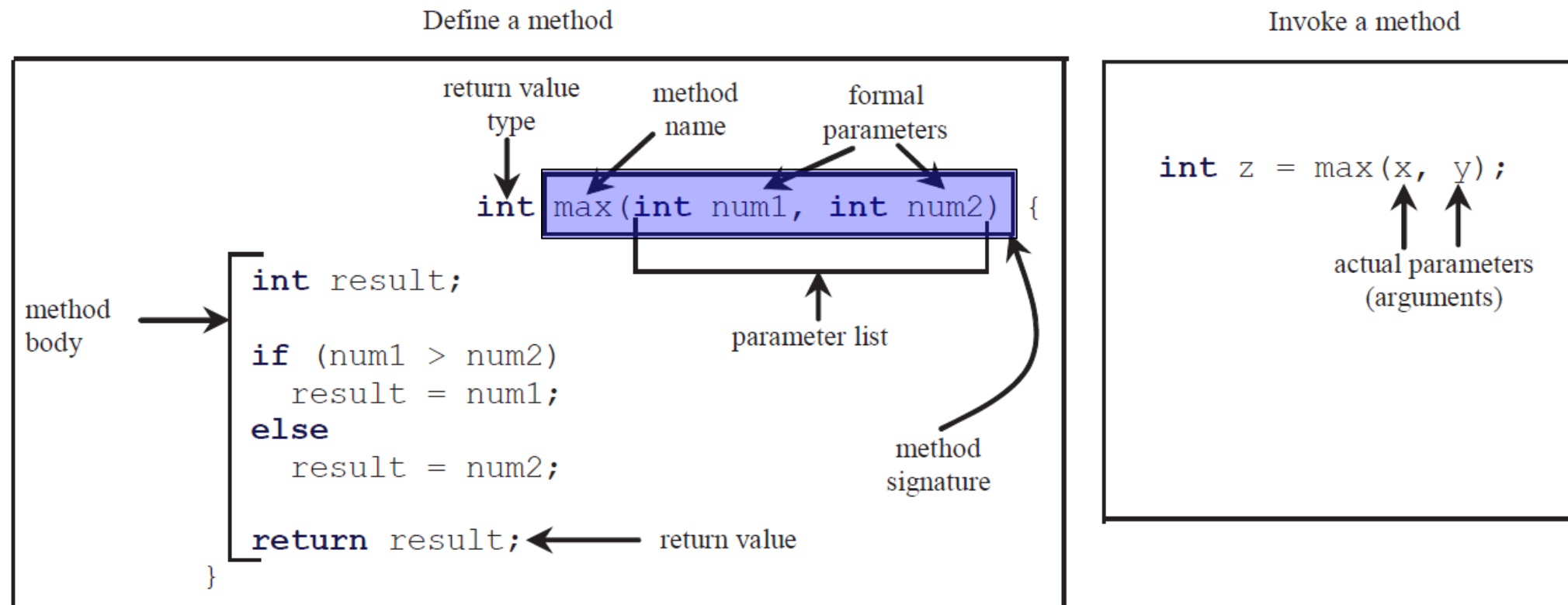
# Defining Methods

- A method is a collection of statements that are grouped together to perform an operation.



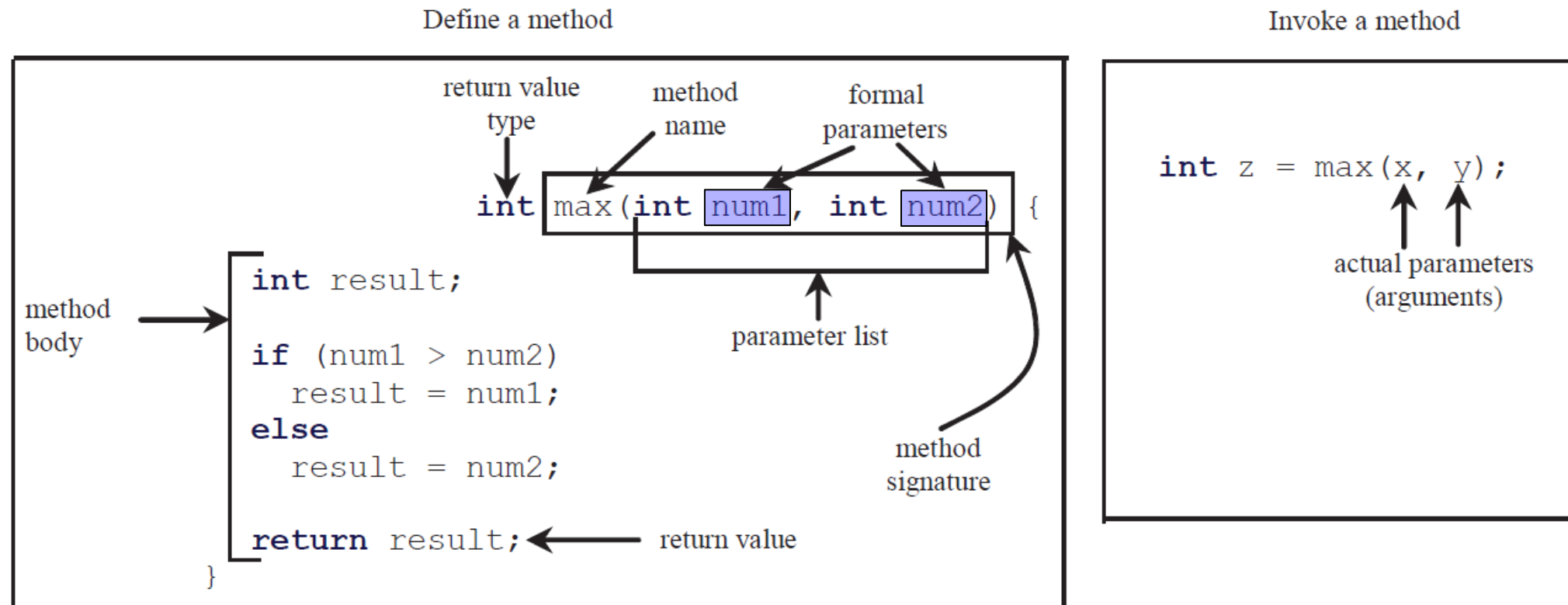
# Method Signature

➤ **Method signature** is the combination of the method name and the parameter list.



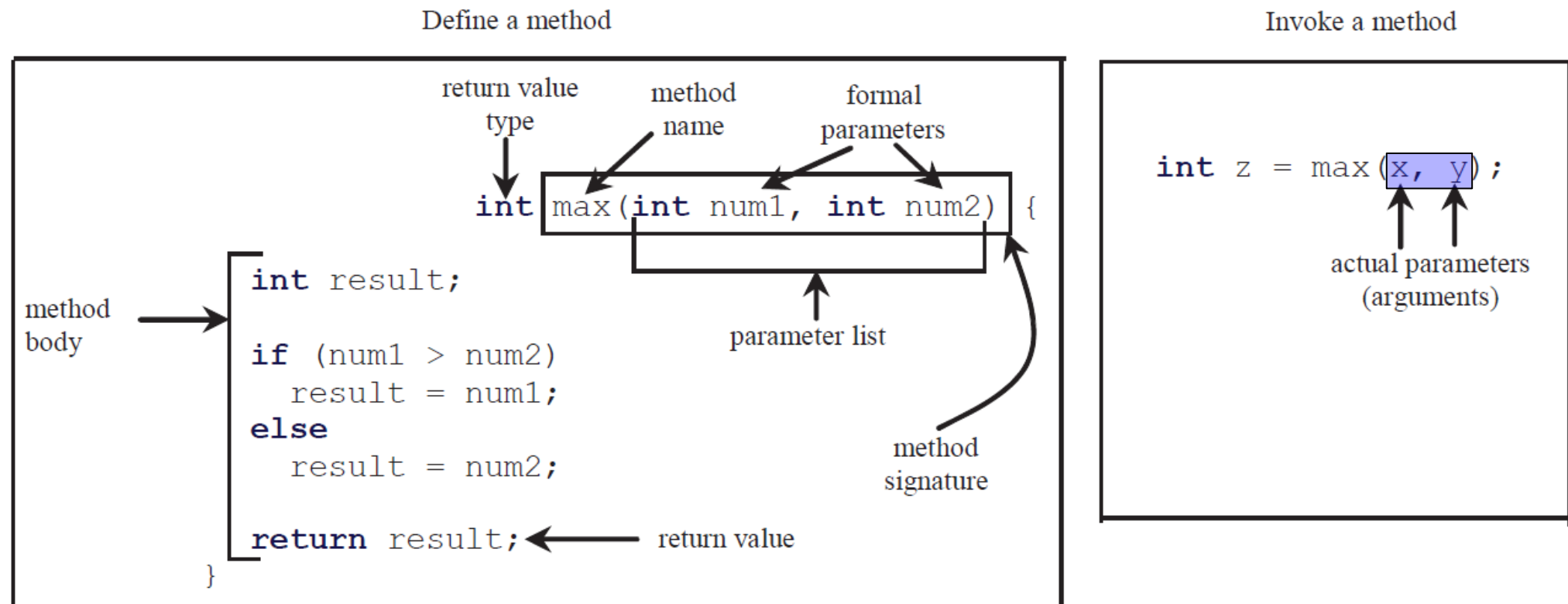
# Formal Parameters

- The variables defined in the method header are known as **formal parameters**.



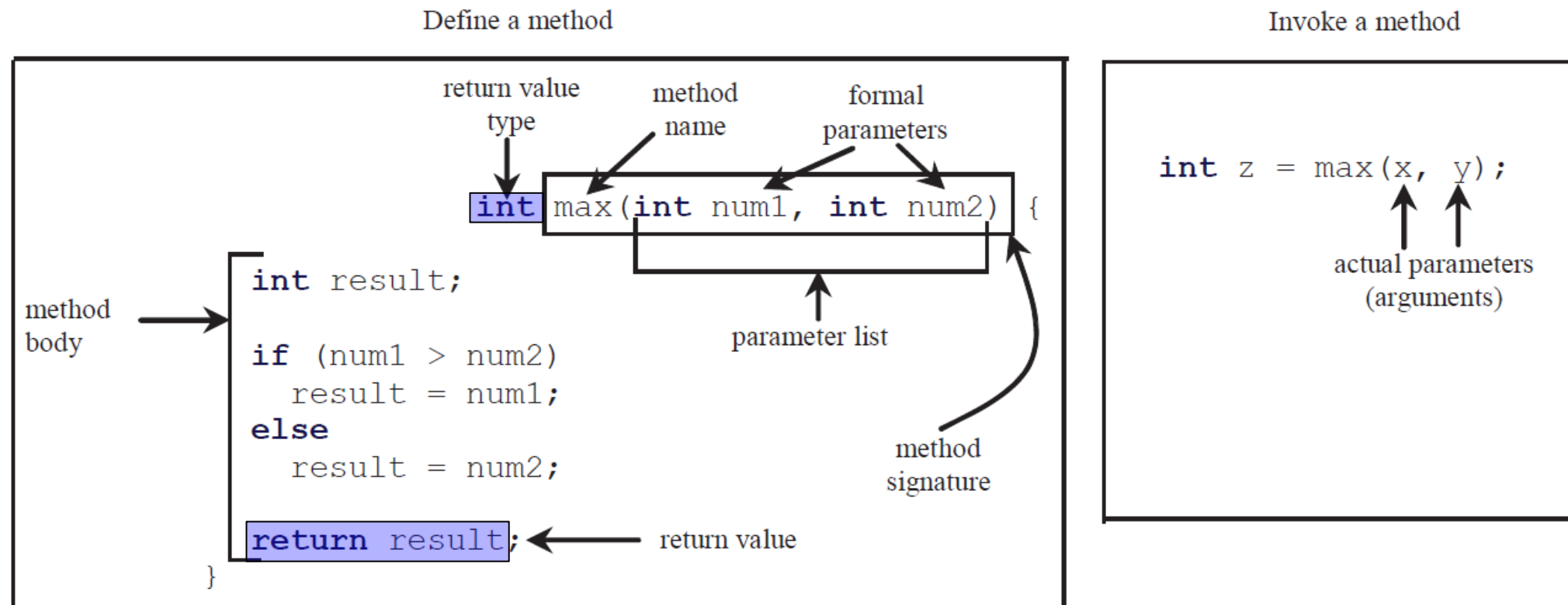
# Actual Parameters

- When a method is invoked, you pass a value to the parameter. This value is referred to as **actual parameter** or **argument**.



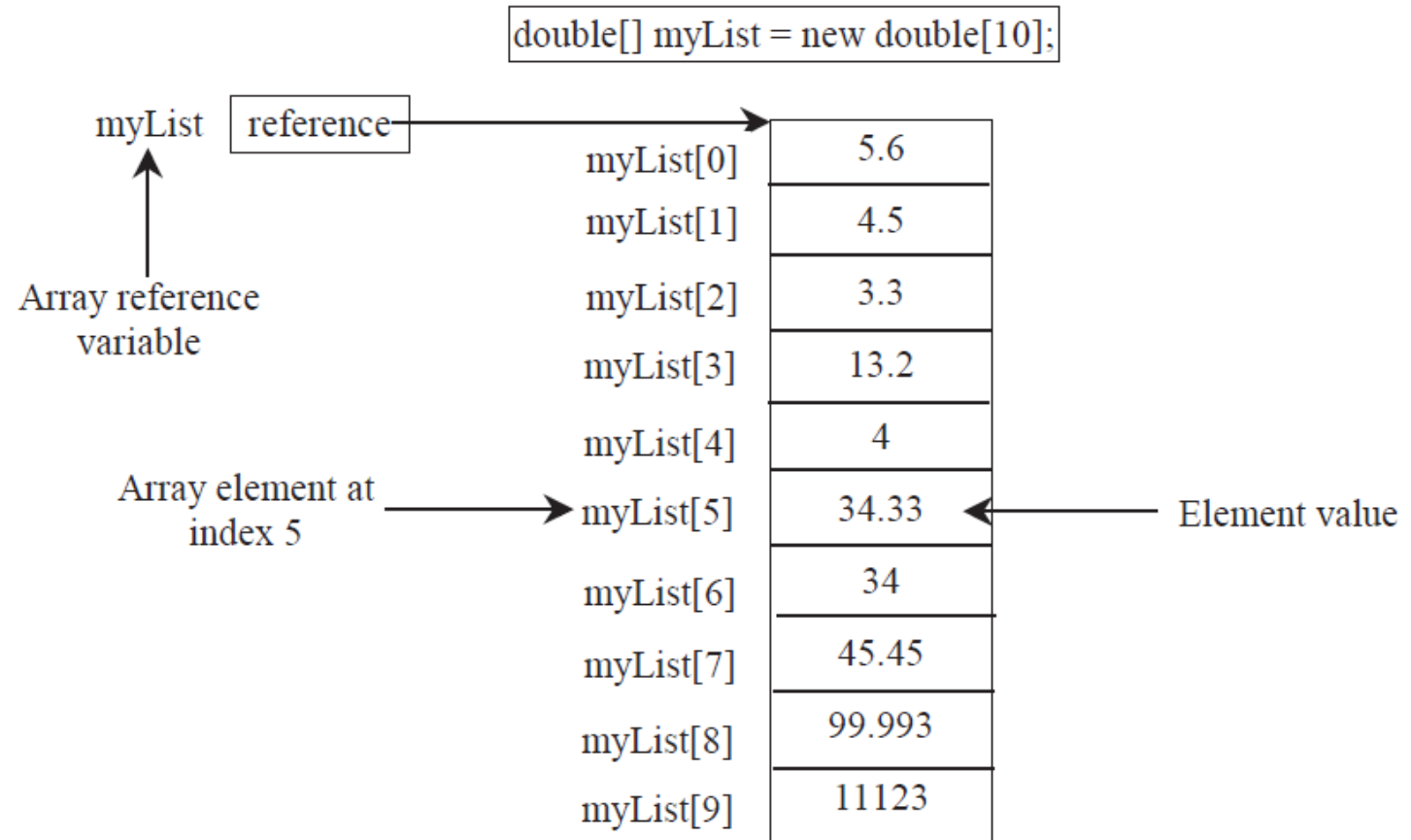
# Return Value Type

- A method may return a value. The returnValueType is the data type of the value the method returns. If the method does not return a value, the returnValueType is the keyword void. For example, the returnValueType in the main method is void.



# Introducing Arrays

- Array is a data structure that represents a collection of the **same types** of data.



# Declaring Array Variables

---

➤ `datatype[] arrayRefVar;`

Example:

`double[] myList;`

➤ `datatype arrayRefVar[];`

Example:

`double myList[];`



# Creating Arrays

---

```
arrayRefVar = new datatype[arraySize];
```

Example:

```
myList = new double[10];
```

`myList[0]` references the first element in the array.

`myList[9]` references the last element in the array.

# Declaring and Creating in One Step

---

➤ `datatype[] arrayRefVar = new  
    datatype[arraySize];`

**Example:** `double[] myList = new double[10];`

➤ `datatype arrayRefVar[] = new  
    datatype[arraySize];`

**Example:** `double myList[] = new double[10];`

# The Length of an Array

---

Once an array is created, its size is **fixed**. It **cannot** be changed.  
You can find its size using

```
arrayRefVar.length
```

Example:

```
myList.length returns 10
```

# Indexed Variables

---

- The array elements are **accessed** through the **index**.
- The array indices are **0-based**, i.e., it starts from **0** to **arrayRefVar.length-1**.
- For example: **myList** holds ten double values and the indices are from 0 to 9.
- Each element in the array is represented using the following syntax, known as an **indexed variable**:

`arrayRefVar[index];`

# Using Indexed Variables

---

- After an array is created, an indexed variable can be used in the same way as a regular variable.
- For example, the following code adds the value in `myList[0]` and `myList[1]` to `myList[2]`.

```
myList[2] = myList[0] + myList[1];
```

# Array Initializers

---

```
double[] myList = new double[4];
```

```
myList[0] = 1.9;
```

```
myList[1] = 2.9;
```

```
myList[2] = 3.4;
```

```
myList[3] = 3.5;
```

# Declaring, Creating, Initializing using the Shorthand Notation

---

- Declaring, creating, initializing in one statement:

```
double[] myList = {1.9, 2.9, 3.4, 3.5};
```

- This shorthand syntax must be in one statement.

# CAUTION

---

- Using the shorthand notation, you have to **declare**, **create**, and **initialize** the array all in one statement. Splitting it would cause a syntax error. For example, the following is **wrong**:

```
double[] myList;
```

```
myList = { 1.9, 2.9, 3.4, 3.5 };
```



# Initializing arrays with input values

---

```
Scanner input = new Scanner(System.in);  
System.out.print("Enter " + myList.length +  
    " values: ");  
for (int i = 0; i < myList.length; i++)  
    myList[i] = input.nextDouble();
```

# Initializing arrays with random values

---

```
for (int i = 0; i < myList.length; i++) {  
    myList[i] = Math.random() * 100;  
}
```

# Printing arrays

---

```
for (int i = 0; i < myList.length; i++) {  
    System.out.print(myList[i] + " ");  
}
```

# Summing all elements

---

```
double total = 0;
for (int i = 0; i < myList.length; i++) {
    total += myList[i];
}
```

# Finding the largest element

---

```
double max = myList[0];  
for (int i = 1; i < myList.length; i++) {  
    if (myList[i] > max) max = myList[i];  
}
```