

Methods

COMP2026

PROBLEM SOLVING USING OBJECT ORIENTED PROGRAMMING

Methods

- ❖ A method is a **sequence of instructions with a name**. We have already encountered several methods. For example, the **Math.pow** method contains instructions to compute a power x^y
- ❖ We call a method in order to execute its instructions. E.g.

```
void runApp()  
{  
    double result = Math.pow(2, 3);  
    ...  
}
```

- ❖ **runApp()** calls the **Math.pow** method to compute 2^3 . The instructions of **Math.pow** method execute and the method **returns** its result back to **runApp**

Method Declaration

Syntax:

```
returnType methodName(parameterType parameterName, ...)  
{  
    method body  
}
```

Type of
return value

Name of
method

Type of parameter
variable

Name of parameter
variable


Method body,
executed
when the
method is
called

```
double squareArea(double sideLength)  
{  
    double area = sideLength * sideLength;  
    return area;  
}
```

return statement exits the
method and the return result

Program Flow

```
10  
11  
13  
14  
15  
16  
17  
18  
19  
20  
21
```

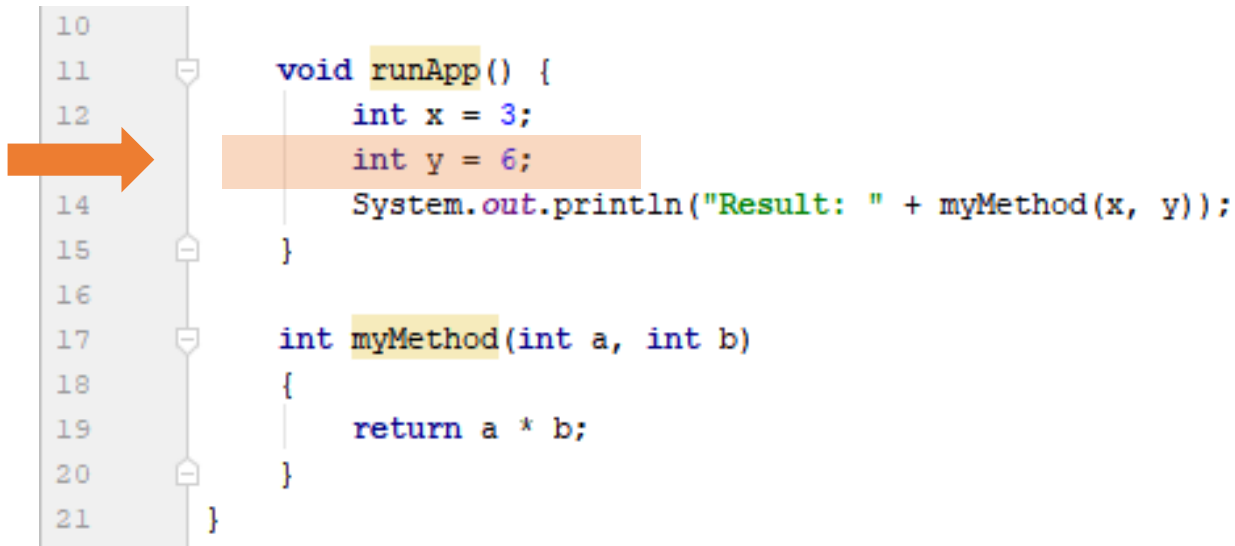


```
void runApp() {  
    int x = 3;  
    int y = 6;  
    System.out.println("Result: " + myMethod(x, y));  
}  
  
int myMethod(int a, int b)  
{  
    return a * b;  
}
```

x = 3

Program Flow

```
10
11 void runApp() {
12     int x = 3;
13     int y = 6;
14     System.out.println("Result: " + myMethod(x, y));
15 }
16
17 int myMethod(int a, int b)
18 {
19     return a * b;
20 }
21 }
```



x = 3
y = 6

Program Flow

```
10
11
12
13
15
16
17
18
19
20
21
```

```
void runApp() {
    int x = 3;
    int y = 6;
    System.out.println("Result: " + myMethod(x, y));
}

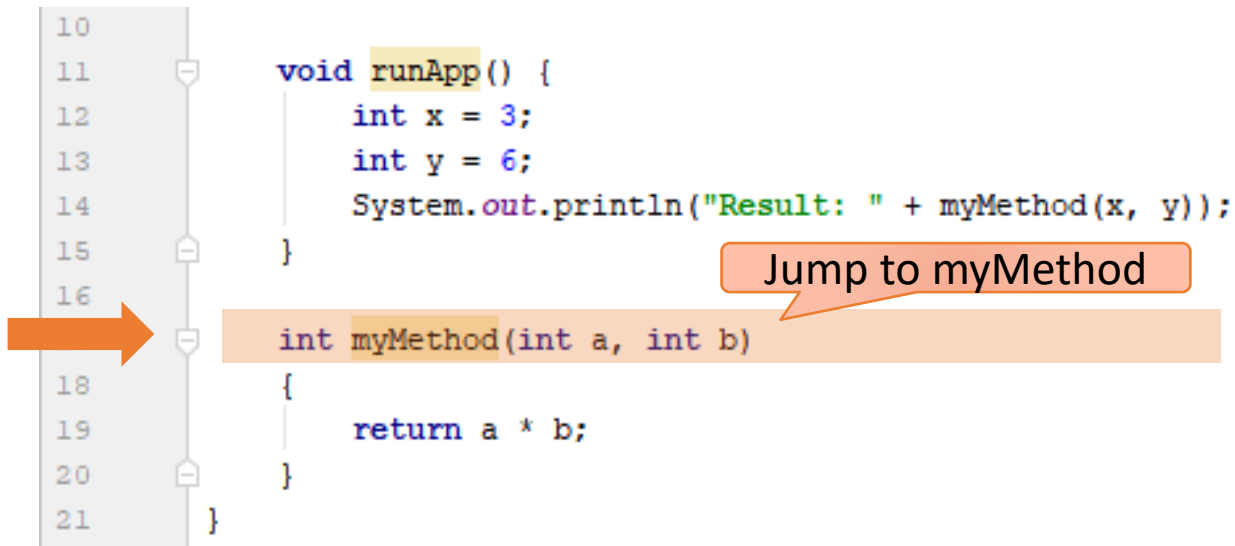
int myMethod(int a, int b)
{
    return a * b;
}
```

Method call

x = 3
y = 6

Program Flow

```
10  
11 void runApp() {  
12     int x = 3;  
13     int y = 6;  
14     System.out.println("Result: " + myMethod(x, y));  
15 }  
16  
17  
18 int myMethod(int a, int b)  
19 {  
20     return a * b;  
21 }
```

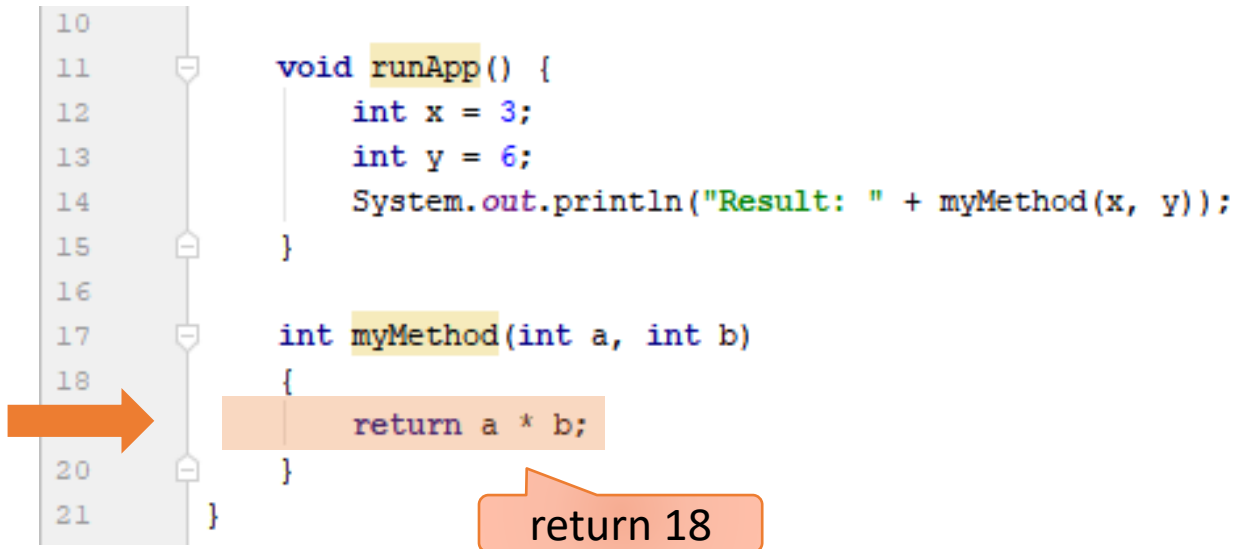


x = 3
y = 6

a = 3
b = 6

Program Flow

```
10
11 void runApp() {
12     int x = 3;
13     int y = 6;
14     System.out.println("Result: " + myMethod(x, y));
15 }
16
17 int myMethod(int a, int b)
18 {
19     return a * b;
20 }
21 }
```

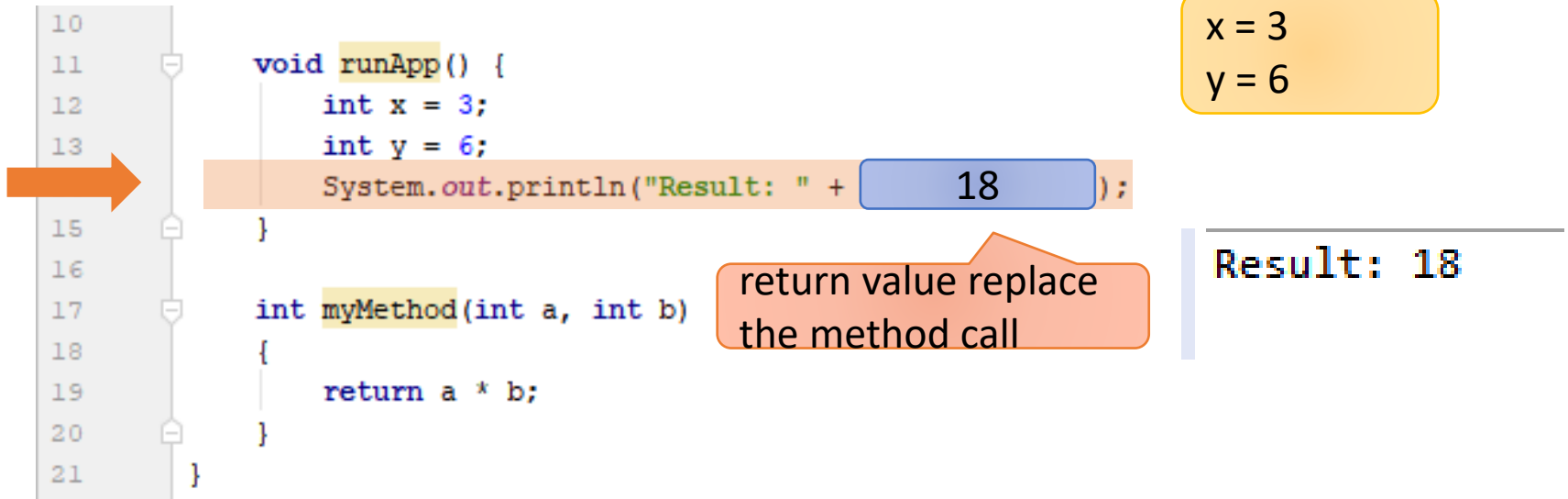


return 18

x = 3
y = 6

a = 3
b = 6

Program Flow



Passing Array to Method

```
void runApp() {  
    int [] x = {12, 52, 23, 47, 56};  
    printArray(x);  
}
```

Passing the array to
method by its name

Type of parameter
variable

Name of parameter
variable

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

Part A

Discovery Exercises

Type your answer in **XXXXXXXXXX_lab06.docx**

Part B

Programming Exercises

Lab Exercise Submission

❖ Submit the following to Moodle

❖ XXXXXXXX_lab06.docx

❖ XXXXXXXX_lab06.zip

*Replace “XXXXXXX” with your student ID

Deadline: Before next Monday noon

References

- ❖ Dean, J., & Dean, R. (2008). *Introduction to programming with Java: A problem solving approach*. Boston: McGraw-Hill.
- ❖ Forouzan, B. A., & Gilberg, R. F. (2007). *Computer science: A structured programming approach using C* (3rd ed.). Boston, MA: Thomson Course Technology.
- ❖ Gaddis, T. (2016). *Starting out with Java* (6th ed.). Pearson.
- ❖ Liang, Y. D. (2013). *Introduction to Java programming: Comprehensive version*. (8th ed.). Pearson.
- ❖ Schildt, H. (2006). *Java a beginner's guide*. New York: McGraw Hill.
- ❖ Wu, C. T. (2010). *An introduction to object-oriented programming with Java*. Boston: McGraw Hill Higher Education
- ❖ Xavier, C. (2011). *Java programming: A practical approach*. New Delhi: Tata McGraw Hill.
- ❖ Zakhour, S., Kannan, S., & Gallardo, R. (2013). *The Java tutorial: A short course on the basics* (5th ed.).
- ❖ yet another insignificant Programming Notes. (n.d.). Retrieved from <https://www3.ntu.edu.sg/home/ehchua/programming>