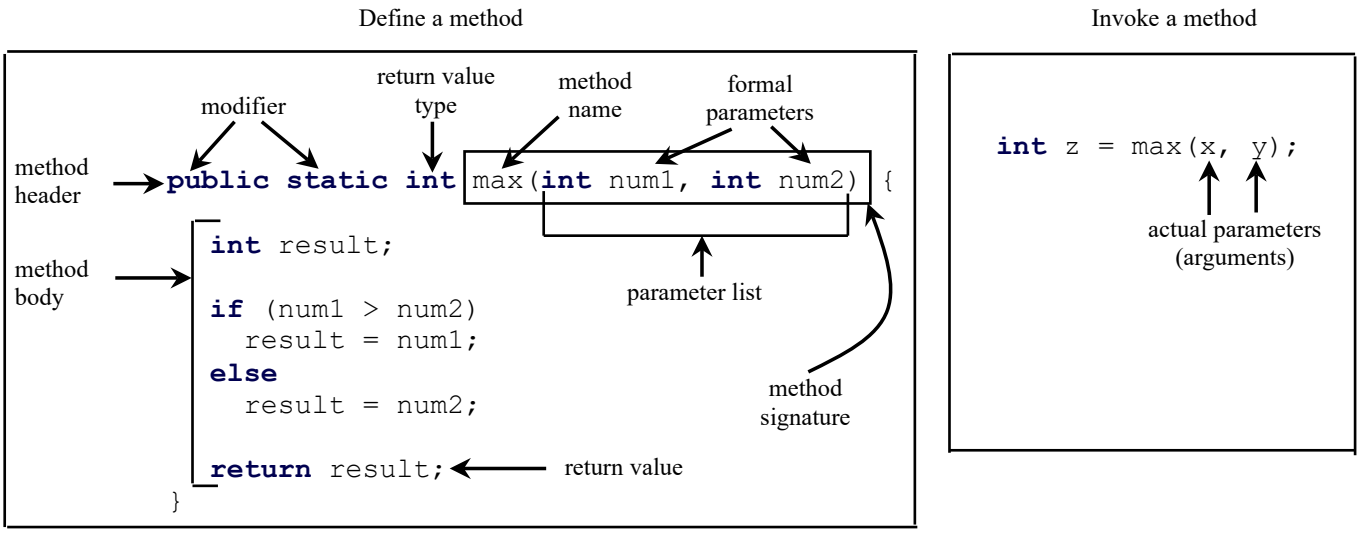
# PKG 10: METHODS 02




***Q1.*** ***In the below examples, line #19 -#108, please work with a classmate to identify the components listed a-j, to trace the data flow, and to explain the output:***

a. Actual parameter(s)
b. Formal parameter(s)
c. Method body
d. Method header
e. Method name
f. Method signature
g. Modifier(s)
h. Parameter list
i. Return value
j. Return value type

```
Method STARTS
Row 0: 0 1 2 3 4
Row 1: 0 1 2 3 4
Row 2: 0 1 2 3 4
Row 3: 0 1 2 3 4
Row 4: 0 1 2 3 4
Method ENDS

Method STARTS
Row 0: 0 1 2 3 4
Row 1: 0 1 2 3 4
Method ENDS

Method STARTS
Row 0: 0 1 2 3 4 5 6 7
Method ENDS

Method STARTS
Row 0: 0 1 2
Row 1: 0 1 2
Row 2: 0 1 2
Row 3: 0 1 2
Row 4: 0 1 2
Method ENDS
```

```
public class LoopsVsMethods02 {

    public static void main(String[] args) {
        traverseBuilding(5, 5);
        traverseBuilding(2, 5);
        traverseBuilding(1, 8);
        traverseBuilding(5, 3);
    }

    public static void traverseBuilding(int floors, int offices) {
        System.out.println("Method STARTS");
        for (int i = 0; i < floors; i++) {
            System.out.print("Row " + i + ": ");
            for (int j = 0; j < offices; j++) {
                System.out.print(j + " ");
            }
            System.out.println("");
        }
        System.out.println("Method ENDS\n");
    }
}
```

---

```
public static void main(String[] args) {
        System.out.println("Happy Coding!");
}
```

```
Happy Coding!
```

## METHODS 02

```
public class MethodDemo {

    public static int sum(int i1, int i2) {
        int result = 0;
        for (int i = i1; i <= i2; i++) {
            result += i;
        }

        return result;
    }

    public static void main(String[] args) {
        System.out.println("Sum from 1 to 10 is " + sum(1, 10));
        System.out.println("Sum from 20 to 37 is " + sum(20, 37));
        System.out.println("Sum from 35 to 49 is " + sum(35, 49));
    }
}
```

```
Sum from 1 to 10 is 55
Sum from 20 to 37 is 513
Sum from 35 to 49 is 630
```

---

```
public class TestReturnGradeMethod {

    public static void main(String[] args) {
        System.out.print("The grade is " + getGrade(78.5));
        System.out.println("");
        System.out.print("The grade is " + getGrade(59.5));
    }

    public static char getGrade(double score) {
        if (score >= 90.0) {
            return 'A';
        } else if (score >= 80.0) {
            return 'B';
        } else if (score >= 70.0) {
            return 'C';
        } else if (score >= 60.0) {
            return 'D';
        } else {
            return 'F';
        }
    }
}
```

```
The grade is C
The grade is F
```

---

```
public class Increment {

    public static void main(String[] args) {
        int x = 1;
        System.out.println("Before the call, x is " + x);
        increment(x);
        System.out.println("After the call, x is " + x);
    }

    public static void increment(int n) {
        n++;
        System.out.println("n inside the method is " + n);
    }
}
```

```
Before the call, x is 1
n inside the method is 2
After the call, x is 1
```

**METHODS 02**

---

***Q2.*** ***Please work with a classmate***

- Outline your solutions to Challenge 1 of PKG 09
- Explain to each in detail the methods you are using

---

***Q3.*** ***Please work with a classmate***

- Discuss ASMT 02 challenges and approaches to solve the challenges
- Discuss the methods which we can use to structure answer to the E.C. part

---

***Q4.*** ***Recode***

- ASMT 01 using methods
- ASMT 02 using methods

---

***Q5.*** ***Please work with a classmate***

1. Suppose your method does not return any value, which of the following keywords can be used as a return type?

a. void b. int c. double d. public e. None of the above

2. The signature of a method consists of _____.

a. method name b. method name and parameter list
c. return type, method name, and parameter list d. parameter list

3. All Java applications must have a method _____.

a. public static Main(String[] args) b. public static Main(String args[])
c. public static void main(String[] args) d. public void main(String[] args)
e. public static main(String[] args)

4. Arguments to methods always appear within _____.

a. brackets b. parentheses c. curly braces d. quotation marks

5. Does the return statement in the following method cause compile errors?

```
public static void main(String[] args) {
  int max = 0;
  if (max != 0)
    System.out.println(max);
  else
    return;
}
```

a. Yes b. No

6. Does the method call in the following method cause compile errors?

```
public static void main(String[] args) {
  Math.pow(2, 4);
}
```

a. Yes b. No

7. Each time a method is invoked, the system stores parameters and local variables in an area of memory, known as _____, which stores elements in last-in first-out fashion.

a. a heap b. storage area c. a stack d. an array

8. Which of the following should be defined as a void method?

a. Write a method that prints integers from 1 to 100.
b. Write a method that returns a random integer from 1 to 100.
c. Write a method that checks whether a number is from 1 to 100.
d. Write a method that converts an uppercase letter to lowercase.

9. You should fill in the blank in the following code with _____.

```
public class Test {
  public static void main(String[] args) {
    System.out.print("The grade is ");
    printGrade(78.5);

    System.out.print("The grade is ");
    printGrade(59.5);
  }

  public static __________ printGrade(double score) {
    if (score >= 90.0) {
      System.out.println('A');
    }
    else if (score >= 80.0) {
      System.out.println('B');
    }
    else if (score >= 70.0) {
      System.out.println('C');
    }
    else if (score >= 60.0) {
      System.out.println('D');
    }
    else {
      System.out.println('F');
    }
  }
}
```

a. int  b. double  c. boolean  d. char  e. void

10. You should fill in the blank in the following code with _____.

```
public class Test {
  public static void main(String[] args) {
    System.out.print("The grade is " + getGrade(78.5));
    System.out.print("\nThe grade is " + getGrade(59.5));
  }

  public static _________ getGrade(double score) {
    if (score >= 90.0)
      return 'A';
    else if (score >= 80.0)
      return 'B';
    else if (score >= 70.0)
      return 'C';
    else if (score >= 60.0)
      return 'D';
    else
      return 'F';
  }
}
```

a. int  b. double  c. boolean  d. char  e. void

11. Consider the following incomplete code:

```
public class Test {
  public static void main(String[] args) {
    System.out.println(f(5));
  }

  public static int f(int number) {
    // Missing body
  }
}
```

The missing method body should be _____.
a. return "number";  b. System.out.println(number);
c. System.out.println("number");  d. return number;

12. When you invoke a method with a parameter, the value of the argument is passed to the parameter. This is referred to as _____.
a. method invocation  b. pass by value  c. pass by reference  d. pass by name

13. Given the following method

```
static void nPrint(String message, int n) {
  while (n > 0) {
    System.out.print(message);
    n--;
  }
}
```

What is the output of the call nPrint('a', 4)?
a. aaaaa  b. aaaa  c. aaa  d. invalid call

14. Given the following method

```
static void nPrint(String message, int n) {
  while (n > 0) {
    System.out.print(message);
    n--;
  }
}
```

What is k after invoking nPrint("A message", k)?

```
int k = 2;
nPrint("A message", k);
```

a. 0
b. 1
c. 2
d. 3

15. Analyze the following code:

```
public class Test {
  public static void main(String[] args) {
    System.out.println(xMethod(5, 500L));
  }

  public static int xMethod(int n, long l) {
    System.out.println("int, long");
    return n;
  }

  public static long xMethod(long n, long l) {
    System.out.println("long, long");
    return n;
  }
}
```

a. The program displays int, long followed by 5.
b. The program displays long, long followed by 5.
c. The program runs fine but displays things other than 5.
d. The program does not compile because the compiler cannot distinguish which method to invoke.

16. Analyze the following code:

```
class Test {
  public static void main(String[] args) {
    System.out.println(xmethod(5));
  }

  public static int xmethod(int n, long t) {
    System.out.println("int");
    return n;
  }

  public static long xmethod(long n) {
    System.out.println("long");
    return n;
  }
}
```

a. The program displays int followed by 5.
b. The program displays long followed by 5.
c. The program runs fine but displays things other than 5.
d. The program does not compile because the compiler cannot distinguish which xmethod to invoke.

17. Analyze the following code.

```
public class Test {
  public static void main(String[] args)  {
    System.out.println(max(1, 2));
  }

  public static double max(int num1, double num2) {
    System.out.println("max(int, double) is invoked");

    if (num1 > num2)
      return num1;
    else
      return num2;
  }

  public static double max(double num1, int num2) {
    System.out.println("max(double, int) is invoked");

    if (num1 > num2)
      return num1;
    else
      return num2;
  }
}
```

a. The program cannot compile because you cannot have the print statement in a non-void method.
b. The program cannot compile because the compiler cannot determine which max method should be invoked.
c. The program runs and prints 2 followed by "max(int, double)" is invoked.
d. The program runs and prints 2 followed by "max(double, int)" is invoked.
e. The program runs and prints "max(int, double) is invoked" followed by 2.

18. A variable defined inside a method is referred to as _____.
a. a global variable  b. a method variable  c. a block variable  d. a local variable

19.  What is k after the following block executes?

```
{
  int k = 2;
  nPrint("A message", k);
}
System.out.println(k);
```