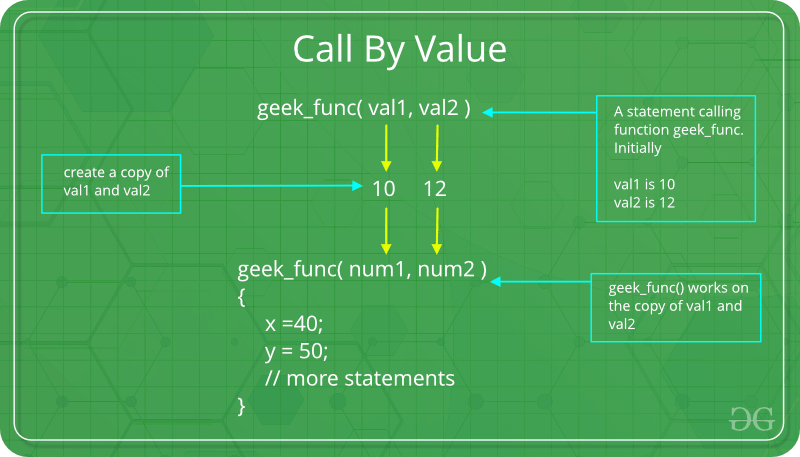
**Passing Variables into Methods**

**Pass By Value:**



// Java program to illustrate

// Call by Value

// Callee

class CallByValue {

// Function to change the value

// of the parameters

public static void Example(int x, int y)

{

x++;

y++;

}

}

// Caller

public class Main {

public static void main(String[] args)

{

int a = 10;

int b = 20;

// Instance of class is created

CallByValue object = new CallByValue();

System.out.println("Value of a: " + a

+ " & b: " + b);

// Passing variables in the class function

object.Example(a, b);

// Displaying values after

// calling the function

System.out.println("Value of a: "

+ a + " & b: " + b);

}

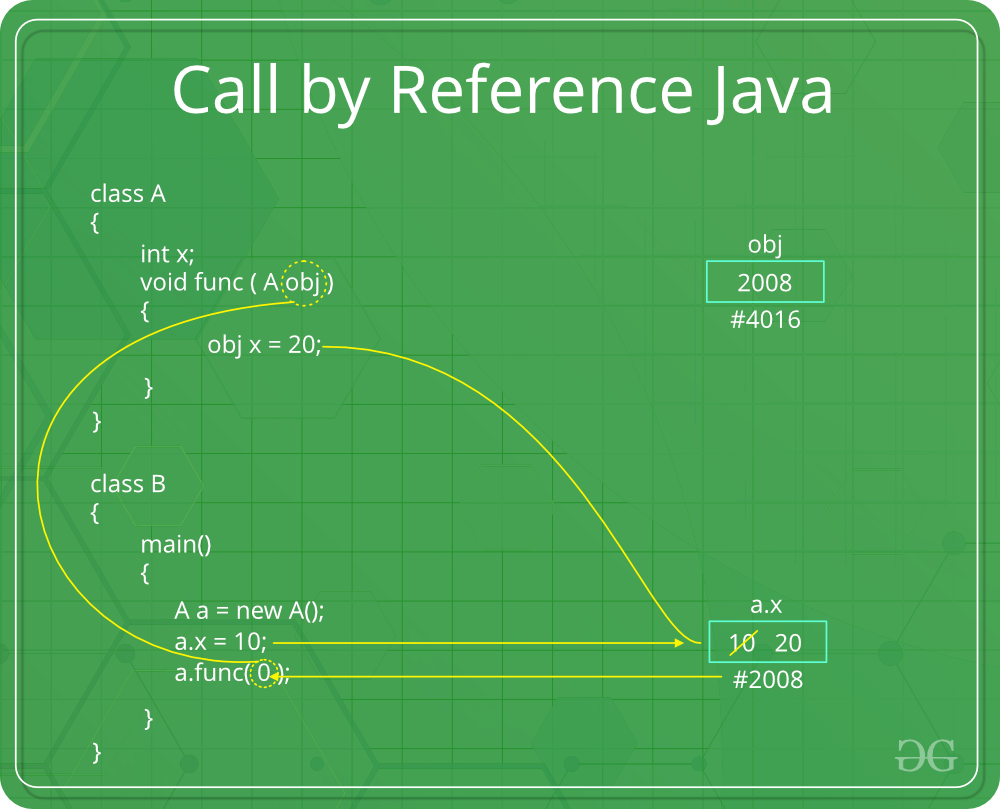
}

**Output:**

Value of a: 10 & b: 20

Value of a: 10 & b: 20

**Call by reference(aliasing)**



// Java program to illustrate

// Call by Reference

// Callee

class CallByReference {

int a, b;

// Function to assign the value

// to the class variables

CallByReference(int x, int y)

{

a = x;

b = y;

}

// Changing the values of class variables

void ChangeValue(CallByReference obj)

{

obj.a += 10;

obj.b += 20;

}

}

// Caller

public class Main {

public static void main(String[] args)

{

// Instance of class is created

// and value is assigned using constructor

CallByReference object

= new CallByReference(10, 20);

System.out.println("Value of a: "

+ object.a

+ " & b: "

+ object.b);

// Changing values in class function

object.ChangeValue(object);

// Displaying values

// after calling the function

System.out.println("Value of a: "

+ object.a

+ " & b: "

+ object.b);

}

}

**Output:**

Value of a: 10 & b: 20

Value of a: 20 & b: 40

## 3. Parameters and Arguments

public class Main {

static void myMethod(String fname) {

System.out.println(fname + " Refsnes");

}

public static void main(String[] args) {

myMethod("Liam");

myMethod("Jenny");

myMethod("Anja");

}

}

O/P:

Liam Refsnes  
Jenny Refsnes  
Anja Refsnes

## 4. Multiple Parameters

## public class Main {

## static void myMethod(String fname) {

## System.out.println(fname + " Refsnes");

## }

## public static void main(String[] args) {

## myMethod("Liam");

## myMethod("Jenny");

## myMethod("Anja");

## }

## }

## O/P:

## Liam Refsnes Jenny Refsnes Anja Refsnes

**MCQ’s**

**1) What is the output of the below Java program with two classes?**

**//Testing1.java**

**public class Example**

**{**

**}**

**public class Testing1**

**{**

**public static void main(String[] args)**

**{**

**System.out.println("Hello Boss.!");**

**}**

**}**

A) Hello Boss.!

B) No Output

C) Compiler error

D) None of the above

Answer [=]

**C**

**Explanation:**

**There can not be more than one public class declared inside a single java file.**

**2) What is the output of the below Java program?**

**//bingo.java file**

**public class Hello**

**{**

**public static void main(String[] args)**

**{**

**System.out.println("BINGO");**

**}**

**}**

A) bingo

B) BINGO

C) Compiler error

D) None

Answer [=]

**C**

**Explanation:**

**The class name and the java file name should be the same. So, change either file name or class name to match.**

**3) State TRUE or FALSE. A Java class provides encapsulation.**

A) TRUE

B) FALSE

C) -

D) -

Answer [=]

**A**

**4) What is the output of the below java class?**

**class Fox**

**{**

**int legs = 2;**

**}**

**class Testing2**

**{**

**public static void main(String[] args)**

**{**

**Fox t1 = new Fox();**

**System.out.println("T1 before: " + t1.legs);**

**t1.legs = 4;**

**System.out.println("T1 After: " + t1.legs);**

**}**

**}**

A)

T1 before: 4

T1 After: 4

B)

T1 before: 2

T1 After: 2

C)

T1 before: 2

T1 After: 4

D) Compiler error

Answer [=]

**C**

**Explanation:**

**There can be any number of classes in a single .java file.**

**5) The value of one primitive variable is assigned to another primitive variable by \_\_\_ in Java.**

A) Pass by value

B) Pass by reference

C) -

D) -

Answer [=]

**A**

**6) A primitive variable is passed from one method to another method by \_\_\_ in Java.**

A) Pass by value

B) Pass by reference

C) -

D) -

Answer [=]

**A**

**7) An object or primitive value that is passed from one method to another method is called \_\_\_ in Java. (Argument / Parameter)**

A) Argument

B) Parameter

C) -

D) -

Answer [=]

**B**

**8) An object or a primitive value that is received in a method from another method is called \_\_\_ in Java. (Argument / Parameter)**

A) Argument

B) Parameter

C) -

D) -

Answer [=]

**A**

**9) What is the output of the below Java program that passes an object to another method?**

**class Food**

**{**

**int items;**

**int show()**

**{return items;}**

**}**

**class Testing9**

**{**

**public static void main(String[] args)**

**{**

**Food f = new Food();**

**f.items = 5;**

**System.out.println("Items Before = " + f.show());**

**change(f);**

**System.out.println("Items After = " + f.show());**

**}**

**static void change(Food foo)**

**{ foo.items = 10; }**

**}**

A)

Items Before = 10

Items After = 10

B)

Items Before = 5

Items After = 5

C)

Items Before = 5

Items After = 10

D)

Items Before = 10

Items After = 5

Answer [=]

**C**

**10) What is the output of the below Java program that passes primitive values?**

**class Testing10**

**{**

**int rats = 5;**

**public static void main(String[] args)**

**{**

**Testing10 t1 = new Testing10();**

**System.out.println("Rats Before = " + t1.rats);**

**modify(t1.rats);**

**System.out.println("Rats After = " + t1.rats);**

**}**

**static void modify(int r)**

**{ r = 20; }**

**}**

A)

Rats Before = 5

Rats After = 5

B)

Rats Before = 20

Rats After = 20

C)

Rats Before = 5

Rats After = 20

D)

Rats Before = 20

Rats After = 5

Answer [=]

**A**

**Explanation:**

**The primitive values are passed by value only. So, changes in the method modify does not change the original value.**