**Passing Object Reference Variables**

**What are Pass-by-value and Pass-by-reference?**

First, let’s understand what are pass-by-value and pass-by-reference.

***Pass-by-value:***

                A copy of the passed-in variable is copied into the argument of the method. Any changes to the argument do not affect the original one.

***Pass-by-reference:***

The argument is an alias of the passed-in variable. Any changes to the argument will affect the original one.

1.

/\*\*

 \* Impossible Swap function in Java

 \* @author www.codejava.net

 \*/

public class Swap {

    public static void swap(int x, int y) {

        int temp = x;

        x = y;

        y = temp;

        System.out.println("x(1) = " + x);

        System.out.println("y(1) = " + y);

    }

    public static void main(String[] args) {

        int x = 10;

        int y = 20;

        swap(x, y);

        System.out.println("x(2) = " + x);

        System.out.println("y(2) = " + y);

    }

}

O/P

x(1) = 20

y(1) = 10

x(2) = 10

y(2) = 20

2.

// Java program to demonstrate reference

// varibale in java

import java.io.\*;

class Demo {

int x = 10;

int display()

{

System.out.println("x = " + x);

return 0;

}

}

class Main {

public static void main(String[] args)

{

Demo D1 = new Demo(); // point 1

System.out.println(D1); // point 2

System.out.println(D1.display()); // point 3

}

}

O/P:

Demo@214c265e

x = 10

0

3.

import java.io.\*;

class Demo {

int x = 10;

int display()

{

System.out.println("x = " + x);

return 0;

}

}

class Main {

public static void main(String[] args)

{

// create instance

Demo D1 = new Demo();

// accessing instance(object) variable

System.out.println(D1.x);

// point 3

// accessing instance(object) method

D1.display();

}

}

O/P:

10

x = 10

4.

// Pointing to same instance memory

import java.io.\*;

class Demo {

int x = 10;

int display()

{

System.out.println("x = " + x);

return 0;

}

}

class Main {

public static void main(String[] args)

{

// create instance

Demo D1 = new Demo();

// point to same reference

Demo G1 = D1;

Demo M1 = new Demo();

Demo Q1 = M1;

// updating the value of x using G!

// reference variable

G1.x = 25;

System.out.println(G1.x); // Point 1

System.out.println(D1.x); // Point 2

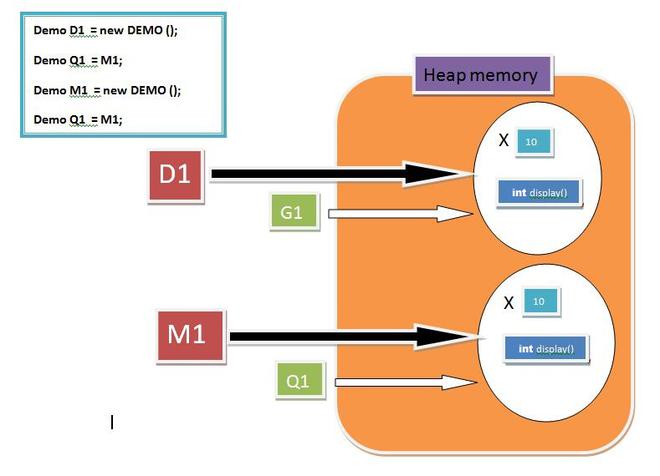
}

}

**Output**

25

25



5.

// Pass by reference and value

import java.io.\*;

class Demo {

int x = 10;

int y = 20;

int display(Demo A, Demo B)

{

// Updating value using argument

A.x = 95;

System.out.println("x = " + x);

System.out.println("y = " + y);

return 0;

}

}

class Main {

public static void main(String[] args)

{

Demo C = new Demo();

Demo D = new Demo();

// updating value using primary reference

// variable

D.y = 55;

C.display(C, D); // POINT 1

D.display(C, D); // POINT 2

}

}

**Output**

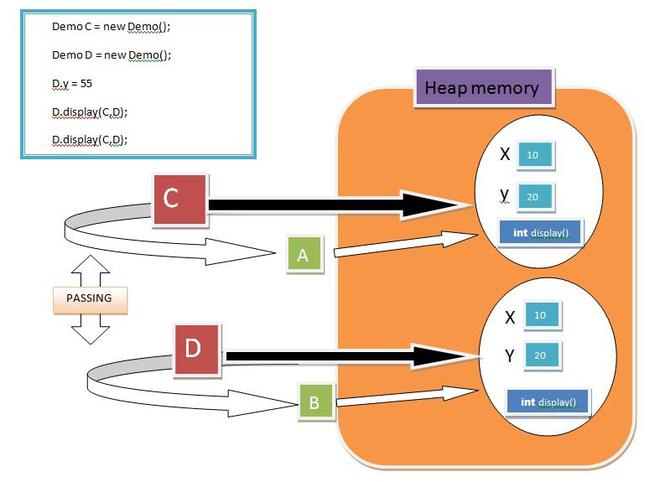
x = 95

y = 20

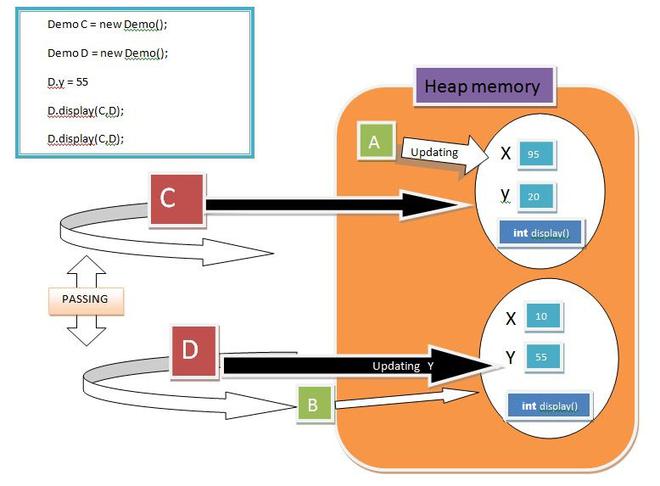
x = 10

y = 55

**SCENE 1 :**



**SCENE 2:**



6.

// Swapping object references

import java.io.\*;

class Demo {

// Swapping Method

int Swap(Demo A, Demo B)

{

Demo temp = A;

A = B;

B = temp;

return 0;

}

}

class Main {

public static void main(String[] args)

{

Demo C = new Demo();

Demo D = new Demo();

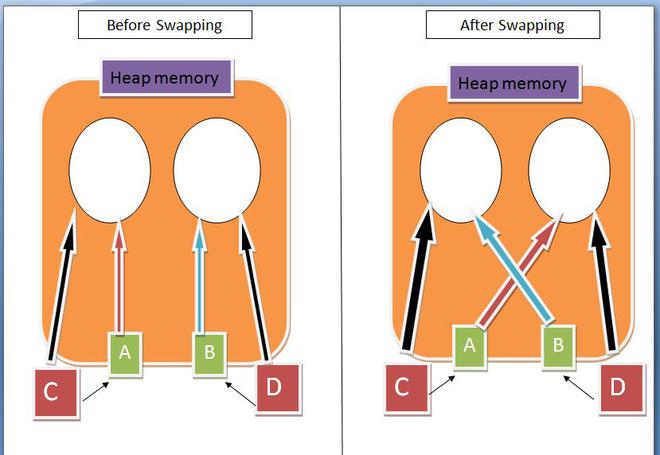
// Passing C and reference variables

// to Swap method

C.Swap(C, D);

}

}



MCQ

**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 [=]

**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 [=]

**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.**

**11) Java object assignment happens by \_\_\_.**

A) Pass by Value

B) Pass by Reference

C) -

D) -

Answer [=]

**B**

**Explanation:**

**Yes. That is the reason why you can change the values of variables of the object using another reference.**

**12) Java object passing from one method to another method happens by \_\_\_.**

A) Pass by Value

B) Pass by Reference

C) -

D) -

Answer [=]

**B**

**Explanation:**

**References point to the original objects. So they can change the state of the objects.**

**13) In Java Pass by reference \_\_\_ is passed even if you are passing a reference to an object.**

A) Address value

B) Variable value

C) Hash code

D) None of the above

Answer [=]

**A**

**Explanation:**

**Yes. The address is passed automatically by Java. So, Java pundits argue that it is passing a value (Address).**

**14) A Java reference is comparable to \_\_\_ in C language.**

A) Enum

B) Structure

C) Pointer

D) None

Answer [=]

**C**

**15) \_\_\_ is the superclass to all Java classes either user-defined or built-in.**

A) Class

B) Object

C) Superclass

D) Null

Answer [=]

**B**

**Explanation:**

**Yes. java.lang.Object is the superclass to all Java classes.**