HEAP and STACK Diagram for Sample.java :

HEAP MEMORY

1000x : SampleObject, i1 : 2000x
2000x : IntegerObject(900)
3000x : SampleObject, i1 : 4000x
4000x : IntegerObject(~~900~~)
5000x : SampleObject, i1 : 6000x
6000x : IntegerObject(~~900~~) 20

Output :
20
4

STACK MEMORY

main_stack

s1 : ~~1000x~~ null
s2 : 3000x
s3 : null

modify_stack

s : ~~3000x~~ ~~5000x~~ null

Program :

```
public class Sample
{
    private Integer i1 = 900;

    public static void main(String[] args)
    {
        Sample s1 = new Sample();

        Sample s2 = new Sample();

       Sample s3 = modify(s2);

        s1 = null;

      //GC [4 objects 1000x,2000x, 5000x and 6000x]

        System.out.println(s2.i1);
    }
  public static Sample modify(Sample s)
    {
      s.i1=9;
      s = new Sample();
      s.i1= 20;
    System.out.println(s.i1);
      s=null;
      return s;
    }
}
```

HEAP and STACK Diagram for Employee.java

HEAP MEMORY

1000x : EmployeeObject, id : ~~400~~ ~~200~~ 500
2000x : EmployeeObject, id :~~400~~ 400
3000x : EmployeeObject, id : ~~400~~ ~~900~~ 500
4000x : EmployeeObject, id : ~~400~~ ~~900~~

Output
400
500
500
500

STACK MEMORY

main_stack

val : 200
e1 : 1000x
e2 : 3000x

update_stack

e : ~~1000x~~ 3000x

switchEmployees_stack

e1 : 3000x
e2 : ~~4000x~~ 4000x
temp : ~~900~~

```
public class Employee
{
  int id = 100;

    public static void main(String[] args)
    {
        int val = 200;

        Employee e1 = new Employee();

        e1.id = val;

        update(e1);

        System.out.println(e1.id);

      Employee e2 = new Employee();

        e2.id = 900;

        switchEmployees(e2,e1); //3000x  1000x

          //GC [2 objects 2000x and 4000x]

            System.out.println(e1.id);
           System.out.println(e2.id);
     }

    public static void update(Employee e)
    {
      e.id = 500;
        e = new Employee();
        e.id = 400;
        System.out.println(e.id);

    }

    public static void switchEmployees(Employee e1, Employee e2)
    {
        int temp = e1.id;
        e1.id = e2.id; //500
        e2 = new Employee();
        e2.id = temp;
    }
}
```

HEAP and STACK Diagram for Test.java

HEAP MEMORY

1000x : TestObject, t : ~~null~~ 3000x val : 100
2000x : TestObject, t : ~~1000x~~ 2000x val : 200
3000x : TestObject, t : ~~1000x~~ 4000x val : 300
4000x : TestObject, t : 2000x val : 400

t2.t = t3;//3000x
t3.t = t4; //4000x
t1.t = t2.t; //3000x
t2.t = t4.t;//2000x

System.out.println(t1.t.val); //300
System.out.println(t2.t.val); //200
System.out.println(t3.t.val); //400
System.out.println(t4.t.val); //200

STACK MEMORY

main_stack

t1 : 1000x

t2 : 2000x

t3 : 3000x

t4 : 4000x

```
public class Test
{
    Test t;
    int val;

    public Test(int val)
    {
        this.val = val;
    }

    public Test(int val, Test t)
    {
        this.val = val;
        this.t = t;
    }

    public static void main(String[] args)
    {
        Test t1 = new Test(100);

        Test t2 = new Test(200,t1);

        Test t3 = new Test(300,t1);

        Test t4 = new Test(400,t2);

        t2.t = t3;
        t3.t = t4;
        t1.t = t2.t;
        t2.t = t4.t;

    System.out.println(t1.t.val);
    System.out.println(t2.t.val);
    System.out.println(t3.t.val);
    System.out.println(t4.t.val);
    }
}
```

Passing an Object reference to the Constructor (Copy Constructor) :

* We can pass an Object reference to the constructor (copy constructor) so we can copy the content of one object to another object.

```
package com.ravi.copy_constructor;

public class Employee
{
    private int employeeId;
    private String employeeName;

    public Employee(int employeeId, String employeeName)
    {
        super();
        this.employeeId = employeeId;
        this.employeeName = employeeName;
    }

    public int getEmployeeId() {
        return employeeId;
    }

    public String getEmployeeName() {
        return employeeName;
    }

}

package com.ravi.copy_constructor;

public class Manager {
    int managerId;
    String managerName;

    public Manager(Employee emp) //emp = e1
    {
        this.managerId = emp.getEmployeeId();
        this.managerName = emp.getEmployeeName();
    }

    @Override
    public String toString()
    {
        return "Manager [managerId=" + managerId + ", managerName=" + managerName + "]";
    }

}

package com.ravi.copy_constructor;

public class CopyConstructor {

    public static void main(String[] args)
    {
        Employee e1 = new Employee(111, "Scott");

        Manager m1 = new Manager(e1);
         System.out.println(m1);
    }

}
```