**Day 13\_Java Assignment**

**1. Problem Description:**

What is the Shallow and Deep Copy of Objects?

public class Person {

private int id;

private String name;

// Getters and Setters

}

Write the below piece of code in the main method and complete this below piece of code.

Person person = new Person();

person.setID(1);

person.setName(“Jerry”);

person shallowCopy = ?

person deepCopy = ?

Complete the above code.

**2. My Solution:**

**Shallow Copy:**

A shallow copy is created when a reference variable is copied into a new reference variable using an assignment operator.

In a shallow copy, both the original and copied objects share the same memory references. Consequently, any modifications made to the values within the object will be reflected in both the original and copied objects.

**Deep Copy:**

A deep copy is created by duplicating all the members of the original object, including any nested objects, and allocating distinct memory locations for each member in the new object. Unlike shallow copy, deep copy ensures complete independence between the original and copied objects. Consequently, modifications made to the values within the deep-copied object do not impact the original object, maintaining isolation and preventing unintended propagation of changes.

**Example:**

**package** shallowDeepCopy;

**public** **class** Person4 **implements** Cloneable{

**private** **int** ID;

**private** String name;

**public** **int** getID() {

**return** ID;

}

**public** **void** setID(**int** iD) {

ID = iD;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**protected** Object clone() **throws** CloneNotSupportedException {

**return** **super**.clone();

}

@Override

**public** String toString() {

**return** "[" + **this**.ID + " " + **this**.name + "]";

}

}

**package** shallowDeepCopy;

**public** **class** MainClass {

**public** **static** **void** main(String[] args) **throws** CloneNotSupportedException{

Person4 person = **new** Person4();

person.setID(1);

person.setName("Jerry");

Person4 shallowCopy = person;

System.***out***.println("Shallow Copy");

System.***out***.println("Before Modification");

System.***out***.println(person);

System.***out***.println(shallowCopy);

shallowCopy.setName("Merry");

System.***out***.println("After Modification");

System.***out***.println(person);

System.***out***.println(shallowCopy);

Person4 deepCopy = (Person4) person.clone();

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

System.***out***.println("Deep Copy");

System.***out***.println("Before Modification");

System.***out***.println(person);

System.***out***.println(deepCopy);

deepCopy.setName("John");

System.***out***.println("After Modification");

System.***out***.println(person);

System.***out***.println(deepCopy);

}

}

**Output:**

Shallow Copy

Before Modification

[1 Jerry]

[1 Jerry]

After Modification

[1 Merry]

[1 Merry]

---------------------------------

Deep Copy

Before Modification

[1 Merry]

[1 Merry]

After Modification

[1 Merry]

[1 John]