**Module 20 - Object Serialization -II**

**1. Module Introduction**

Object Serialization-II

Learning Objectives

**In this module you will learn**

Java Transient Keyword

practice more problems on Object serialization and deserialization

Introduction

Serialization is the process of writing the state of an object to a byte stream. This is useful when you want to save the state of your program to persistent data storage like files. Later you can restore these objects by using the process of deserialization.

**2. Resources**

Java Transient Keyword

Java transient keyword is used in serialization. If you define any data member as transient, it will not be serialized.

Let's take an example, I have declared a class as Student, it has three data members id, name and age. If you serialize the object, all the values will be serialized but I don't want to serialize one value, e.g. age then we can declare the age data member as transient.

**Example of Java Transient Keyword**

In this example, we have created the two classes Student and PersistExample. The age data member of the Student class is declared as transient, its value will not be serialized.

If you deserialize the object, you will get the default value for transient variable.

Let's create a class with transient variable.

[](http://wiki2016.msitprogram.net/index.php/File:T.PNG)

Now write the code to serialize the object.

[](http://wiki2016.msitprogram.net/index.php/File:T1.PNG)

**Output**

success

**Now write the code for deserialization.**

[](http://wiki2016.msitprogram.net/index.php/File:T2.PNG)

211 ravi 0

Example to Write Number of records into file using Serialization with Vector

import java.io.\*;

import java.util.\*;

class Employee implements java.io.Serializable

{

public String name;

public String address;

public transient int SSN;

public int number;

public void mailCheck()

{

System.out.println("Mailing a check to " + name

+ " " + address);

}

}

public class Vector\_Example

{

public static void main(String [] args)

{

Employee e = new Employee();

Employee e1 = new Employee();

Vector<Employee> v=new Vector<Employee>();

e.name = "Reyan Ali";

e.address = "Phokka Kuan, Ambehta Peer";

e.SSN = 11122333;

e.number = 101;

e1.name = "Renuka";

e1.address = "Dharmavaram";

e1.SSN = 111;

e1.number = 101;

try

{

FileOutputStream fileOut =

new FileOutputStream("employee.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

v.add(e);

v.add(e1);

out.writeObject(v);

out.close();

fileOut.close();

System.out.printf("Serialized data is saved in employee.txt");

}catch(IOException i)

{

i.printStackTrace();

}

}

}

Example to Read Number of records from file Using Deserialization with Vector

import java.io.\*;

import java.util.\*;

class Employee implements java.io.Serializable

{

public String name;

public String address;

public transient int SSN;

public int number;

public void mailCheck()

{

System.out.println("Mailing a check to " + name

+ " " + address);

}

}

public class Vector\_reading

{

public static void main(String [] args)

{

try

{

FileInputStream fileIn = new FileInputStream("employee.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

Vector<Employee> v=new Vector<Employee>();

v=(Vector<Employee>)in.readObject();

Employee e =new Employee();

for(int i=0;i<v.size();i++)

{

e=v.get(i);

System.out.println("Deserialized Employee...");

System.out.println("Name: " + e.name);

System.out.println("Address: " + e.address);

System.out.println("SSN: " + e.SSN);

System.out.println("Number: " + e.number);

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

}

in.close();

fileIn.close();

}catch(IOException i)

{

i.printStackTrace();

return;

}catch(ClassNotFoundException c)

{

System.out.println("Employee class not found");

c.printStackTrace();

return;

}

}

}