**Serialization-**

The process of storing the state of objects into file called as serialization and process of reading the state of objects from file called as Deserialization.

1. The process of saving (or) writing state of an object to a file is called serialization

2. but strictly speaking it is the process of converting an object from java supported form to either network supported form (or) file supported form.

3. By using FileOutputStream and ObjectOutputStream classes we can achieve serialization process.

**De-Serialization:**

1. The process of reading state of an object from a file is called DeSerialization

2. but strictly speaking it is the process of converting an object from file supported form (or) network supported form to java supported form.

3. By using FileInputStream and ObjectInputStream classes we can achieve DeSerialization

**How to implement serialization in java**

By using the input and output stream, we can do it.

**import** java.io.FileInputStream;

**import** java.io.FileOutputStream;

**import** java.io.ObjectInputStream;

**import** java.io.ObjectOutputStream;

**import** java.io.Serializable;

**class** SerializableTest **implements** Serializable {

**int** i = 10;

**int** j = 20;

}

**public** **class** Demo {

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

SerializableTest d1 = **new** SerializableTest();

System.***out***.println("Serialization started");

FileOutputStream fos = **new** FileOutputStream("C:\\Users\\HP\\OneDrive\\Desktop\\DemoT.txt");

ObjectOutputStream oos = **new** ObjectOutputStream(fos);

oos.writeObject(d1);

System.***out***.println("Serialization ended");

System.***out***.println("Deserialization started");

FileInputStream fis = **new** FileInputStream("C:\\Users\\HP\\OneDrive\\Desktop\\DemoT.txt");

ObjectInputStream ois = **new** ObjectInputStream(fis);

SerializableTest d2 = (SerializableTest) ois.readObject();

System.***out***.println("Deserialization ended");

System.***out***.println(d2.i + "................" + d2.j);

}

}

Output:

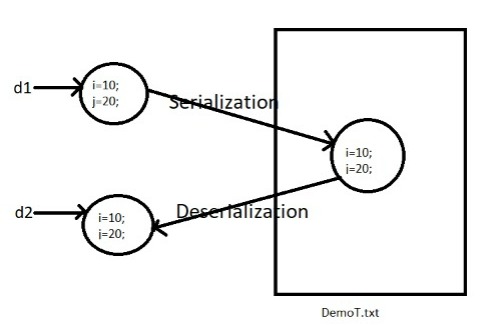
Serialization started

Serialization ended

Deserialization started

Deserialization ended

10................20



Example- Suppose I have one student class in which first name, last name and mobile number. I just want to store that into file name. Then go for serialization.

**package** com.test;

**import** java.io.Serializable;

**public** **class** Student **implements** Serializable {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

String firstname;

String lastname;

String city;

**public** String getFirstname() {

**return** firstname;

}

**public** **void** setFirstname(String firstname) {

**this**.firstname = firstname;

}

**public** String getLastname() {

**return** lastname;

}

**public** **void** setLastname(String lastname) {

**this**.lastname = lastname;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

}

**package** com.test;

**import** java.io.FileOutputStream;

**import** java.io.ObjectOutputStream;

**public** **class** SerializeStudent {

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

Student s = **new** Student();

s.setFirstname("ajay");

s.setLastname("pawar");

s.setCity("pune");

**try** {

FileOutputStream fos = **new** FileOutputStream("C:\\Users\\ThisPC\\Desktop\\demo.txt");

ObjectOutputStream oos = **new** ObjectOutputStream(fos);

oos.writeObject(s);

fos.close();

oos.close();

System.***out***.println("Serialization is done...");

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**package** com.test;

**import** java.io.FileInputStream;

**import** java.io.ObjectInputStream;

**public** **class** DeserializeStudent {

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

**try** {

FileInputStream fis = **new** FileInputStream("C:\\Users\\ThisPC\\Desktop\\demo.txt");

ObjectInputStream ois = **new** ObjectInputStream(fis);

Object o = ois.readObject(); // Read the object

Student s = (Student) o;// convert to student

System.***out***.println(s.getFirstname());

System.***out***.println(s.getLastname());

System.***out***.println(s.getCity());

fis.close();

ois.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

ajay

pawar

pune

**Transient keyword:**

1. transient is the modifier applicable only for variables.

2. While performing serialization if we don't want to save the value of a particular variable to meet security constant such type of variable, then we should declare that variable with "transient" keyword.

3. At the time of serialization JVM ignores the original value of transient variable and save default value to the file.

4. That is transient means "not to serialize".

**Static Vs Transient:**

static variable is not part of object state hence they won't participate in serialization because of this declaring a static variable as transient there is no use.

**Transient Vs Final:**

final variables will be participated into serialization directly by their values. Hence declaring a final variable as transient there is no use. //the compiler assign the value to final variable.

**Serial Version UID:**

* The serialization associated with each serializable class has a version number called Serial Version UID.
* It is used during de-serialization to verify that the sender and receiver of a serialized object have loaded classes for that and are compatible with respect to serialization.
* If the receiver is loaded with different version of a class that has different serial version UIDs than the corresponding sender's class, then de-serialization will result in an invalid Class Exception.
* A Serializable class can declare its own serial version UID explicitly by declaring a field named serial version UID that must be static, final and of type long.

e.g. **private** **static** **final** **long** ***serialVersionUID*** = 1L;

Example -Consider the above same program in which we don't want to serialize the salary of a student

**package** com.test;

**import** java.io.Serializable;

**public** **class** Student **implements** Serializable {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

String firstname;

String lastname;

**transient** **int** salary;

**public** String getFirstname() {

**return** firstname;

}

**public** **void** setFirstname(String firstname) {

**this**.firstname = firstname;

}

**public** String getLastname() {

**return** lastname;

}

**public** **void** setLastname(String lastname) {

**this**.lastname = lastname;

}

**public** **int** getSalary() {

**return** salary;

}

**public** **void** setSalary(**int** salary) {

**this**.salary = salary;

}

}

**package** com.test;

**import** java.io.FileOutputStream;

**import** java.io.ObjectOutputStream;

**public** **class** SerializeStudent {

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

Student s = **new** Student();

s.setFirstname("ajay");

s.setLastname("pawar");

s.setSalary(5000); //wont be serialized

**try** {

FileOutputStream fos = **new** FileOutputStream("C:\\Users\\ThisPC\\Desktop\\demo.txt");

ObjectOutputStream oos = **new** ObjectOutputStream(fos);

oos.writeObject(s);

fos.close();

oos.close();

System.***out***.println("Serialization is done...");

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

**package** com.test;

**import** java.io.FileInputStream;

**import** java.io.ObjectInputStream;

**public** **class** DeserializeStudent {

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

**try** {

FileInputStream fis = **new** FileInputStream("C:\\Users\\ThisPC\\Desktop\\demo.txt");

ObjectInputStream ois = **new** ObjectInputStream(fis);

Object o = ois.readObject(); // Read the object

Student s = (Student) o;// convert to student

System.***out***.println(s.getFirstname());

System.***out***.println(s.getLastname());

System.***out***.println(s.getSalary());

fis.close();

ois.close();

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

Output:

ajay

pawar

0