**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.

**How to implement serialization in java**

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

Example-1 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

* The ObjectOutputStream and ObjectInputStream are used to serialize and de-serialize objects respectively.
* If we don't want to serialize some fields of class then we use the transient keyword. If any member is declared as transient then it won't be serialized.
* If the superclass implements serializable interface, then all its subclasses will be serializable by default.
* All static members of class are not serialized because static members are related to class only, not to object.
* 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.
* If a superclass variable is made transient, then after de-serialization, it gives default value like zero or null.

Example 2-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