Ex No: 1 Illustrate the concept of Serialization and Deserialization using File

AIM:

To implement the Serialization and Deserialization using File

ALGORITHM:

Step 1: Creating a Student class and serializing the object of the Student class.

Step 2: Serialization of an Object of type Student. A text file called f.txt  is created with the help of the FileOutputStream class. Serializing the object by using the writeObject() method of ObjectOutputStream class.

Step 3: For deserializing the object by using the readObject() method of ObjectInputStream class.

**PROGRAM:**

**Student.java**

**import** java.io.Serializable;

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

**int** rno;

  String name;

 float fees;

**public** Student(**int** id, String name, float fees) {

**this**.rno = id;

**this**.name = name1;

this.fees = fees1;

 }

public String toString() {

return rno + “ “ + name + “ “ + fees + “\n”;

}

**Serial.java**

**import** java.io.\*;

**class** Serial{

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

**try**

{

  Student s1 =**new** Student(1,"ram", 1000);

    FileOutputStream fout=**new** FileOutputStream("d:\\f.txt");

   ObjectOutputStream out=**new** ObjectOutputStream(fout);

   out.writeObject(s1);

   out.flush();

   out.close();

   System.out.println("success");

  }

**catch**(Exception e){

System.out.println(e);

}

 }

}

**Deserial.java**

**import** java.io.\*;

**class** Deserial {

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

**try**{

   FileInputStream fin =  **new** FileInputStream("d:\\f.txt")

   ObjectInputStream in=**new** ObjectInputStream(**fin**);

   Student s=(Student)in.readObject();

    System.out.println(s);

   in.close();

  }**catch**(Exception e){

System.out.println(e);

}

 }

}

RESULT:

Thus the Serialization and Deserialization using File was implemented in a Java console application.