Serialization

- **present in java.io package**

- Mechanism of writing the state of an object into a byte-stream.
- Mainly used to travel object's state on the network (known as marshaling)
- For serializing the object, call the **writeObject()** method of **ObjectOutputStream** *class*.
- The class must implement **Serializable** interface in order to serialize the object. Serializable is a **marker interface.**

Eg:-

```
class Student implements Serializable {
        public int rollNo;
        public String name;

        Student(int rollNo, String name) {
                this.rollNo = rollNo;
                this.name = name;
        }
}

public class SerializationExample {
        pvsm(..) throws IOException {
                Student s = new Student(1,"Tushar","DIT");
                FileOutputStream fos = new FOS(“abc.text");
                ObjectOutputStream oos =  new OOS(fos);
                oos.writeObject(s);
        }
}
```

- ***If a class implements serializable then all its sub classes will also be serializable.***

- ***If a class has a reference to another class, all the references must be Serializable otherwise serialization process will not be performed. In such case, NotSerializableException is thrown at runtime.***

- ***If there is any static data member in a class, it will not be serialized because static member is the part of class not object.***

– ***Any data member marked with transient keyword will not be serialized. So when this file is deserialized the value for that particular data member will be set to default(0 for integer, null for String and so on).***

SerialVersionUID

Deserialization

- Reverse of serialisation, i.e. byte-stream is converted into object.
- For deserialization call the **readObject()** method of **ObjectInputStream** class.

Eg:-

```
public class DeserializationExample {
        pvsm(..) throws ClassNotFoundException {
                FileInputStream fis = new FileInputStream(“abc.text");
                ObjectInputStream ois = new ObjectInputStream(fis);
                Student s1 = (Student)ois.readObject();
                System.out.println(s1.toString());
        }
}
```