**Module 19 - Object Serialization**

**1. Module Introduction**

Learning Objectives

In this module you will learn

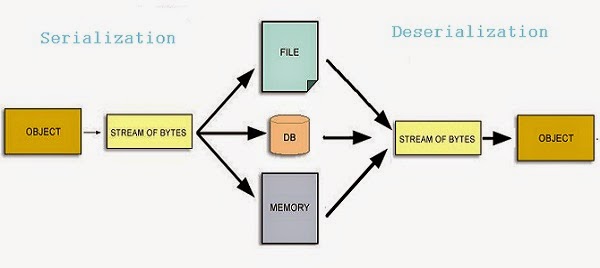
* How to serialize an object
* How to deserialize an object
* How to Use ObjectInputStream
* How to Use ObjectOutputStream

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.

Resources

Process of Serialization and Deserialization Has been explained in below image

[](http://wiki2016.msitprogram.net/index.php/File:SD.JPG)

### 2. Resources

## Video Resources

**Serialization**

<https://youtu.be/6MisF1sxBTo>

**How to Serialize and deSerialize an Objects**

<https://youtu.be/YzwiuRDgSSY>

## Advantage of Java Serialization

It is mainly used to travel object's state on the network (known as marshaling).

## java.io.Serializable interface

Serializable is a marker interface (has no body). It is just used to "mark" java classes which support a certain capability.

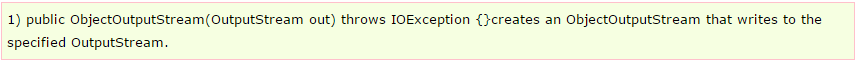
It must be implemented by the class whose object you want to persist. Let's see the example given below:

[750](http://wiki2016.msitprogram.net/index.php?title=Special:Upload&wpDestFile=S1.PNG)

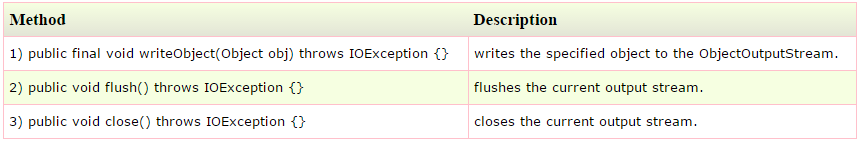
## ObjectOutputStream class

The ObjectOutputStream class is used to write primitive data types and Java objects to an OutputStream. Only objects that support the java.io.Serializable interface can be written to streams.

**Constructor**

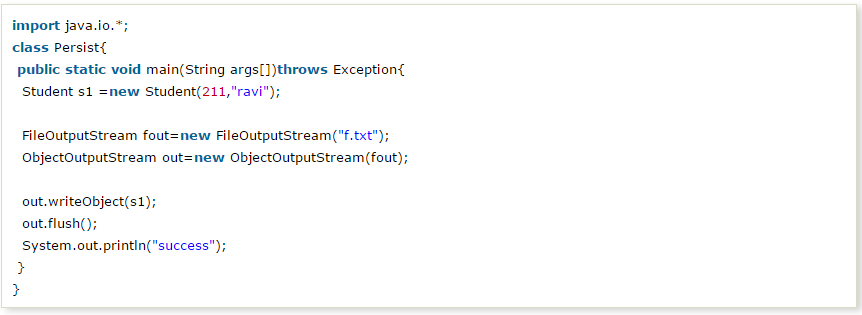
[](http://wiki2016.msitprogram.net/index.php/File:OB1.PNG)

**Important Methods**

[](http://wiki2016.msitprogram.net/index.php/File:OB2.PNG)

**Example of Java Serialization**

In this example, we are going to serialize the object of Student class. The writeObject() method of ObjectOutputStream class provides the functionality to serialize the object. We are saving the state of the object in the file named f.txt.

[](http://wiki2016.msitprogram.net/index.php/File:OB1_1.PNG)

**output**

success...

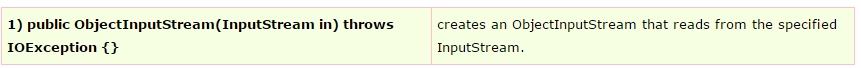
## Deserialization in java

Deserialization is the process of reconstructing the object from the serialized state.It is the reverse operation of serialization.

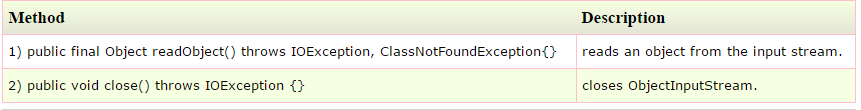
**ObjectInputStream class**

An ObjectInputStream deserializes objects and primitive data written using an ObjectOutputStream.

**Constructor**

[](http://wiki2016.msitprogram.net/index.php/File:OB1_2.PNG)

**Important Methods**

[](http://wiki2016.msitprogram.net/index.php/File:OB1_3.PNG)

**Example of Java Deserialization**

[](http://wiki2016.msitprogram.net/index.php/File:OB1-4.PNG)

## Steps involved in Object Serialization

Let us remind the basic things you must do when you are making a class serializable. They are:

Step 1: Implement the Serializable interface.

import java.io.Serializable;

class SerializationDemo implements Serializable

{

public int marks = 94;

public String Grade=”A+”

}

In step 1, the only thing you need to do is to implement serializable interface to your class.

The serializable interface is a marker interface; it has no methods in it. This interface makes the Java class eligible for serialization.

Step 2: Calling write object.

public static void main(String args[]) throws IOException

{

FileOutputStream fos = new FileOutputStream("temp.txt");

ObjectOutputStream oos = new ObjectOutputStream(fos);

SerializationDemo sd = new SerializationDemo();

oos.writeObject(sd);

oos.flush();

oos.close();

}

The step 2, shows how to store the state of Serialization Demo object in a file called temp.txt

Step 3: Recreating a serialized object.

public static void main(String args[]) throws IOException

{

FileInputStream fis = new FileInputStream("temp.txt");

ObjectInputStream oin = new ObjectInputStream(fis);

SerializationDemo sd = (SerializationDemo) oin.readObject();

System.out.println("Marks="+sd.value);

System.out.println("Grade="+sd.Grade);

}