Lab 8a. // SerializationDeSerialDemo

```
import java.io.*;
import java.io.Serializable;
//Class Student implements
class Student implements Serializable{
    int id;
    String name;
    public Student(int id, String name) {
        this.id = id;
        this.name = name;
    }
}

class SerializationDeSerialDemo{
    public static void main(String args[]){
        try{
            //Create the object of student class
            Student s1 =new Student(27,"James Gosling");
            //Write the object to the stream by creating a output stream
            FileOutputStream fout=new FileOutputStream("James.txt");
            ObjectOutputStream out=new ObjectOutputStream(fout);
            out.writeObject(s1);
            out.flush();
            //close the stream
            out.close();
            System.out.println("Object successfully written to the file");

            //Create a stream to read the object
            ObjectInputStream in=new ObjectInputStream(new FileInputStream("James.txt"));
            Student s=(Student)in.readObject();
            //print the data of the deserialized object
            System.out.println("Student object: " + s.id+" "+s.name);
            //close the stream
            in.close();

        }catch(Exception e){System.out.println(e);}
    }
}
```

Output:

```
Object successfully written to the file
Student object: 27 James Gosling
```

Lab 8b.

```
import java.io.Serializable;
//Class Student implements
class Student1 implements Serializable{
    String name;
    String email;
    transient String password;
    public Student1(String name, String email, String password) {
        this.name = name;
        this.email = email;
        this.password=password;
    }
}

public class BasicSerializationExample {
    static final String filePath = "Student.txt";

    static void serialize(Student1 user) {
        try {
            FileOutputStream fos = new FileOutputStream(filePath);
            ObjectOutputStream outputStream = new ObjectOutputStream(fos);
            outputStream.writeObject(user);
            outputStream.close();
        } catch (IOException ex) {
            System.err.println(ex);
        }
    }

    static Student1 deserialize() {
        Student1 savedUser = null;

        try {
            FileInputStream fis = new FileInputStream(filePath);
            ObjectInputStream inputStream = new ObjectInputStream(fis);
            savedUser = (Student1) inputStream.readObject();
            inputStream.close();
        } catch (IOException | ClassNotFoundException ex) {
            System.err.println(ex);
        }

        return savedUser;
    }

    public static void main(String[] args) {
        String name = "James";
        String email = "info@codejava.net";
        String password = "secret";
        Student1 newUser = new Student1(name, email, password);

        serialize(newUser);

        Student1 s = deserialize();
        //print the data of the deserialized object
        System.out.println("Student object: " + s.name+" "+s.email+s.password);
        //close the stream
    }
}
```

```
Student object: James info@codejava.net
null
```

#Q.) Serialization of Static and Transient Variables

Any static and transient variables declared inside a class cannot be serialized. For example:

```
static int x = 30;

transient String str = "myPassword";
```

These variables cannot be stored in a file. Static is the part of class, not object. So: Null for static and transient.