

## Task 2: Serialization and Deserialization

Serialize a custom object to a file and then deserialize it back to recover the object state.

```
package test;
```

```
import java.io.*;
```

```
class Person implements Serializable {
```

```
    private static final long serialVersionUID = 1L;
```

```
    private String name;
```

```
    private int age;
```

```
    public Person(String name, int age) {
```

```
        this.name = name;
```

```
        this.age = age;
```

```
    }
```

```
    public String getName() {
```

```
        return name;
```

```
    }
```

```
    public int getAge() {
```

```
        return age;
```

```
    }
```

```
    @Override
```

```
    public String toString() {
```

```
        return "Person{" +
```

```
            "name=\"" + name + "\" +
```

```
            ", age=" + age +
```

```
            '}';
```

```
    }
```

```
}
```

```
public class SerializationExample {
```

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

```
        Person p=new Person("Alice", 30);
```

```
        String f="person.ser";
```

```
        serializeObject(p,f);
```

```
        Person deserializedPerson = deserializeObject(f);
```

```

System.out.println("Deserialized Person: "+deserializedPerson);
}

private static void serializeObject(Person object, String filename) {
try (ObjectOutputStream oos=new ObjectOutputStream(new FileOutputStream(filename))) {
oos.writeObject(object);
System.out.println("Object serialized successfully to " + filename);
} catch (IOException e) {
e.printStackTrace();
}
}

private static Person deserializeObject(String filename) {
Person object=null;
try (ObjectInputStream ois=new ObjectInputStream(new FileInputStream(filename))) {
object=(Person)ois.readObject();
System.out.println("Object deserialized successfully from "+filename);
} catch (IOException | ClassNotFoundException e) {
e.printStackTrace();
}
return object;
}
}

```