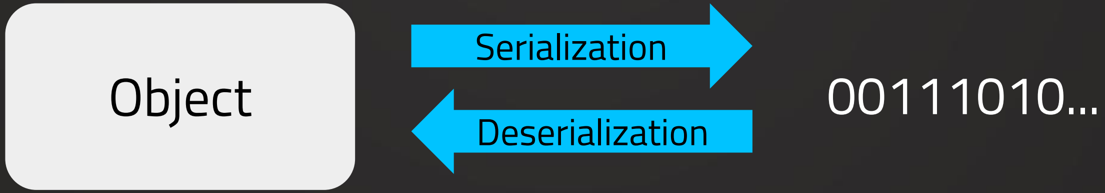
Java Serialization & Deserialization

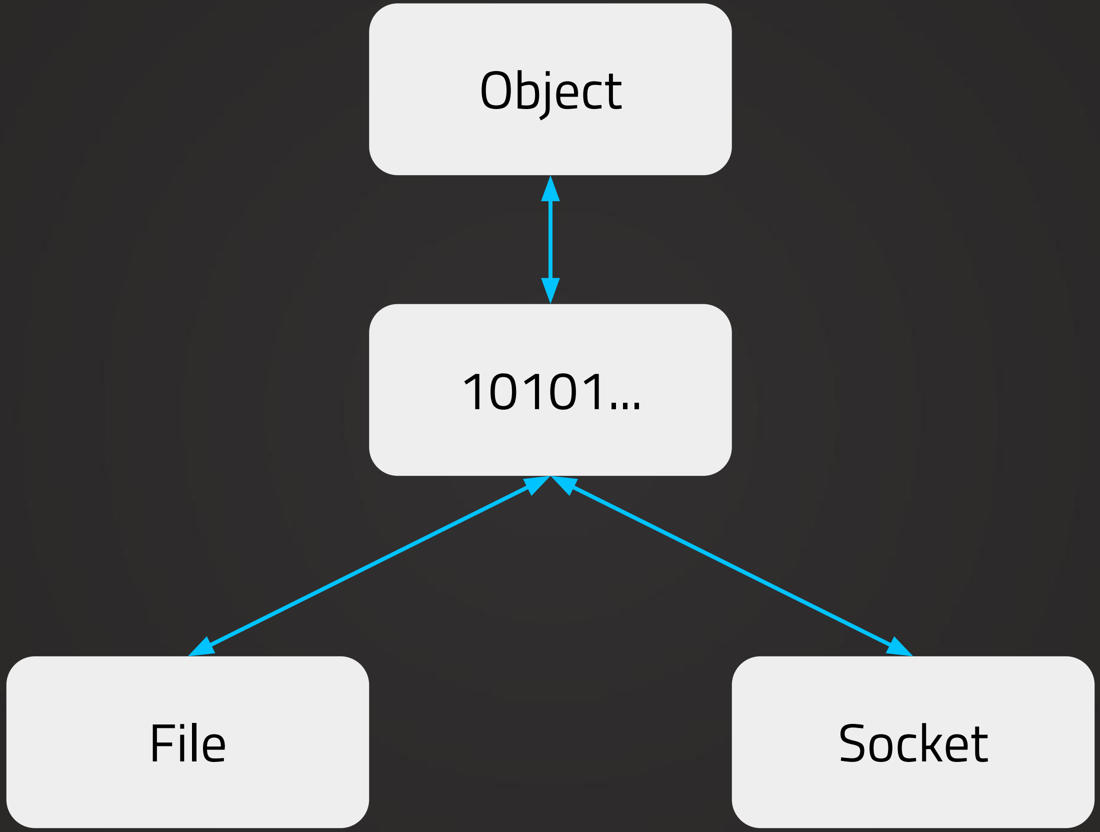
CSCI 4960

Owen Xie

What is Serialization / Deserialization?

- Turn arbitrary objects into a byte stream and back
- Can be done manually or automatically
- Many languages provide automatic mapping
  - Java Serializable
  - Python Pickle




Object
10101...
File
Socket

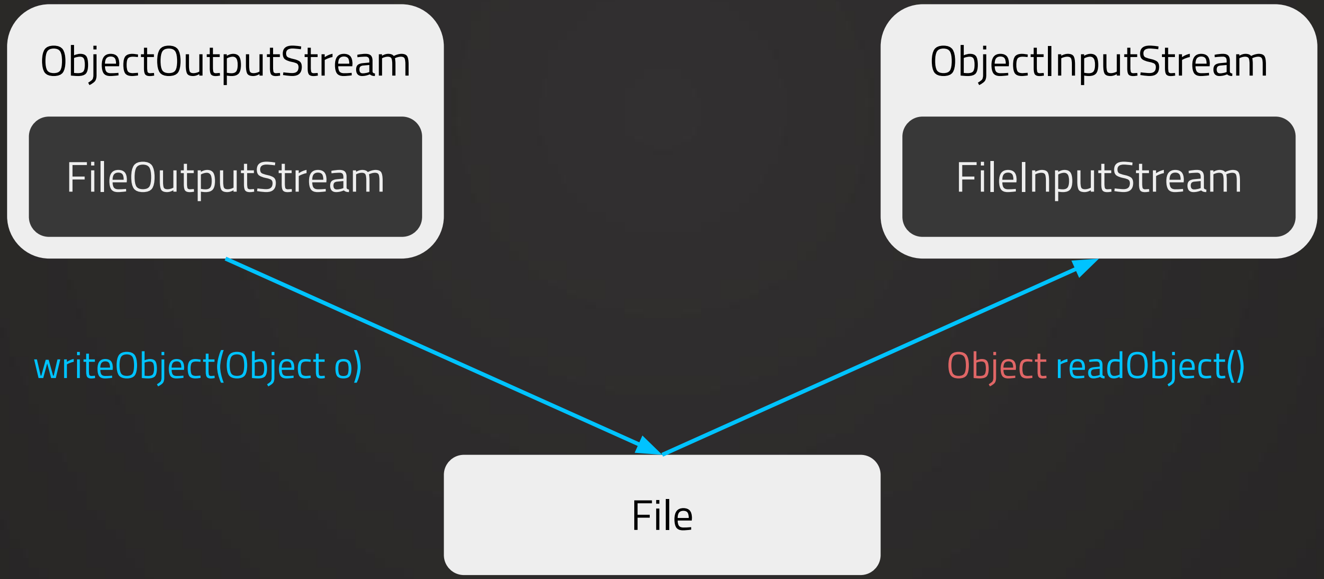
Java Serialization

Java provides automatic serialization if a class implements the Serializable interface.

- No overridden methods needed
- Carries over to subclasses
- All fields must also be serializable or marked “transient”
- Most collections in Java are serializable

Object Streams

Java uses ObjectInputStream and ObjectOutputStream to serialize and deserialize objects.



Customized Serialization

While it is not necessary to implement any methods, you can optionally define:

- `writeObject(ObjectOutputStream os)`
- `readObject(ObjectInputStream is)`
- `writeReplace()`
- `readResolve()`

Serializable Example

```
class Person implements Serializable {
    public String name;
    private Address address;
    private ArrayList<Person> family;
}

class Address implements Serializable {
    // Street, City, Country, Zip Code fields
}
```

Transient?

```
// You can control what is serialized or not
class Login implements Serializable {
    public String username;
    private String password_hash;

    // Don't save plaintext passwords to a file!
    private transient String password;
}
```

Why Use Serialization?

- Reduces the amount of code needed
  - Useful for writing network protocols
- Working with object models is more natural than parsing data from a string

serialVersionUID

Used by the JVM to ensure that an object being read is the same as the object that was written.

- Can be automatically generated
- Will be different on different machines
  - Necessary if sending objects over the network

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

Limitations (In My Experience)

ObjectInputStream expects to take in a “magic header” first

- Peer to peer networking (UDP Sockets) is difficult
    - Needs to separate the magic headers between clients
    - Alternatively create a new stream on each packet
    - Generally not a good experience

Closing Points

Java serialization is great for reducing the amount of boilerplate.

Many alternative libraries for serialization:

- Kyro
- protobuf
- GSON
- fastjson