

# Session 6

Lecturer

Mouhsen Ibrahim

# Contents

- Java IO Streams
- Byte Streams
- Character Streams
- Creating RMI Programs
- Running RMI Programs
- RMI Example Program with chat app

# Java IO

- A stream can be defined as a sequence of data.
- Four kinds of streams:
  - Input Stream: Read data from a source.
  - Output Stream: Write data to a destination.
  - Byte Streams: `InputStream` and `OutputStream` are abstract classes that describe a byte input and output stream.
  - Character Streams: `Reader` and `Writer` are abstract classes that describe character input and output streams (two bytes).

# Byte Streams

- **BufferedInputStream** and **BufferedOutputStream** are used to read and write buffered data.
- **DataInputStream** and **DataOutputStream** are used to read and write java standard data types.
- **FileInputStream** and **FileOutputStream** are used to read/write data from/to a file.
- **PrintStream** is an output stream that contain **print()** and **println()** methods,

# Character Streams

- **BufferedReader** and **BufferedWriter** handle buffered input and output.
- **FileReader** and **FileWriter** read and write characters to/from a file.
- **InputStreamReader** convert bytes to characters.
- **OutputStreamWriter** convert characters to bytes.
- **PrintWriter** is a class that contain `print()` and `println()` methods.

# IO Examples

- The source code is in the IOExample.java file, it contains the following tasks
- Read a character from Console.
- Read a string from console
- Use BufferedReader to read from a file
- Use FileReader to read from a file
- Use FileWriter to write to a file

# Creating RMI Program

- **Step 1:** Create the remote interface, this interface should extend `java.rmi.Remote` and declare all methods that can be called remotely.
- **Step 2:** Create the remote class, this class should implement the remote interface and extend `java.rmi.server.UnicastRemoteObject`, it should implement all remote methods.
- **Step 3:** Write the RMI server that will bind the remote objects to the registry.
- **Step 4:** Write the RMI client that will call remote methods using the remote interface and RMI registry.

# Running RMI Program

- First start the RMI registry by executing `start rmiregistry` command in the bin directory of the server's project.
- Run the server program.
- Take the Interface from the server project to the client's project.
- Run the client program.
- Hint: You can create the RMI registry in the server program using `LocateRegistry.createRegistry(1099);`.
- This will start rmiregistry on default port (1099).

# RMI Example Program with chat app

- Write an RMI Server with one remote method called `sayHello` this method returns “Hello from RMI Server”.
- Write an RMI Client that will call the previous method.
- Convert the previous Chat app to use RMI, implement these methods:
  - Register method: This method takes username and password as arguments and creates a file based on the username with the password inside it.
  - Login method: This method takes username and password as arguments and checks if they are correct then return true or false.

**GOOD LUCK**