

**(Established under the Presidency University Act, 2013 of the Karnataka Act 41 of 2013)**

## CSA2022 – Advanced JAVA

## LAB SHEET – 8

**Module 2 – Input Output Operations in Java**

**Operations with Channel and Buffer**

Buffers work with the channel. Channels are the tube through which data is transferred and buffers are the source and target of those data transfers. In the case of a write, data we want to write is placed in a buffer, which is passed to a channel then the channel reads that data from the buffer and writes it into the file.

**Q4**: Write a Java program to perform read and write operation with a data file using FileChannel and

ByteBuffer.

**Solution :**

**import** java.io.\*;

**import** java.nio.\*;

**import** java.nio.channels.FileChannel;

**public** **class** FileChannelDemo {

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

String data="iphone 6 50000";

*write*("tablet.store", data);

*read*("tablet.store");

}

**public** **static** **void** write(String filename, String data) {

**try** {

RandomAccessFile store = **new** RandomAccessFile(filename, "rw");

FileChannel channel = store.getChannel();

ByteBuffer buffer = ByteBuffer.*allocate*(data.length());

**for** (**int** i = 0; i < data.length(); i++) {

buffer.put((**byte**) (data.charAt(i)));

}

// Rewinds buffer, the position is set to zero

buffer.rewind();

channel.write(buffer);

channel.close();

store.close();

}**catch**(Exception e) {}

}

**public** **static** **void** read(String filename) {

**try** {

RandomAccessFile store = **new** RandomAccessFile(filename, "rw");

FileChannel channel = store.getChannel();

ByteBuffer buffer = ByteBuffer.*allocate*(1024);

**int** numOfBytesRead = channel.read(buffer);

System.***out***.println("number of bytes read : " + numOfBytesRead);

**for**(**int** i=0;i<numOfBytesRead;i++) {

**char** c=(**char**)buffer.get(i);

System.***out***.print(c);

}

channel.close();

store.close();

}**catch**(Exception e) {}

}

}

**Operations with Serialization**

**Q5**: Write a java program to serialize and deserialize a student object under the file name file named student.ser.

**Solution :**

**import** java.io.\*;

**import** java.io.Serializable;

**class** Student **implements** Serializable {

String name;

String regdno;

**double** cgpa;

}

**public** **class** SerializeDemo {

**public** **static** **void** main(String [] args) **throws** Exception {

Student s = **new** Student();

s.name = "sanjay";

s.regdno = "20213cse0123";

s.cgpa = 7.5;

FileOutputStream fileOut =**new** FileOutputStream("tmp//student.ser");

ObjectOutputStream out = **new** ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.***out***.printf("Serialized data is saved in /tmp/employee.ser");

}

}

**import** java.io.\*;

**public** **class** DeserializeDemo {

**public** **static** **void** main(String [] args) **throws** Exception {

Student s = **null**;

FileInputStream fileIn = **new** FileInputStream("tmp//student.ser");

ObjectInputStream in = **new** ObjectInputStream(fileIn);

s = (Student) in.readObject();

in.close();

fileIn.close();

System.***out***.println("Deserialized Student...");

System.***out***.println("Name: " + s.name);

System.***out***.println("Regd no: " + s.regdno);

System.***out***.println("CGPA: " + s.cgpa);

}

}