**Lab 20**

**Random Access Files**

Download the zip file lab20Skeleton.zip from Moodle.

Extract the file and open in NetBeans as a project. You are provided with some of the classes for a student filing system using RAF. In this exercise, you are provided with the code for the additional classes that you must add to the project. Take your time to ensure that you understand the code that you are using. Read the comments in the files provided.

Open the file TestStudentRAF.java. You will see that some lines of code are commented out which call on static methods in classes that are not written yet. You need to create these classes and add in the relevant code in the order shown below.

1. Create a new class called WriteRAFRecord and add the following code into it:

import java.io.\*;

// write 1 record out to RAF file

public class WriteRAFRecord {

public static void writeRecord(File file, int studentNum, String firstName, String lastName, double gpaValue) {

RAFStudentRecord record = new RAFStudentRecord();

try {

// Open file for writing

try (RandomAccessFile raf = new RandomAccessFile(file, "rw")) {

// set the values on the Record class

// the record will be written to file

record.setSnum(studentNum);

record.setFirstName(firstName);

record.setLastName(lastName);

record.setGpa(gpaValue);

// seek moves the file position pointer for object output

// based on calculation below

// Student numbers entered should be between 1 and 100

raf.seek((studentNum - 1) \* RAFStudentRecord.SIZE);

record.write(raf);

}

} catch (FileNotFoundException fnf) {

System.out.println("Error:"+fnf.getMessage());

} catch (IOException io) {

System.out.println("Error"+io.getMessage());

}

}

}

Comment out the call to the writeRecord method in the test class.

1. Create a new class ReadRAFRecord and add in the following code:

import java.io.\*;

// reads 1 record from file

public class ReadRAFRecord {

public static void readRecord(File file, int recordNum) {

RAFStudentRecord record = new RAFStudentRecord();

try {

try (RandomAccessFile raf = new RandomAccessFile(file, "r")) {

// seek moves the file position pointer for object output

// based on calculation below

// Student numbers entered should be between 1 and 100

raf.seek((recordNum - 1) \* RAFStudentRecord.SIZE);

record.read(raf);

// check that there is data in the record

if (record.getSnum() == 0) {

System.out.println("ERROR: That record does not exist");

return;

}

String values[] = {String.valueOf(record.getSnum()),

record.getFirstName(),record.getLastName(),

String.valueOf(record.getGpa())};

for (String value : values) {

System.out.println(value);

}

}

} catch (EOFException eof) {

System.out.println("End of file reached");

} catch (IOException io) {

System.out.println(io.getMessage());

}

}

}

Comment out the call to the readRecord method in the test class.

1. Create a class UpdateRAFRecord and add in the following code:

package solution;

import java.io.\*;

public class UpdateRAFRecord {

public static void updateRecord(File file, int recordNum, String firstName, String lastName, double gpaValue) {

RAFStudentRecord record = new RAFStudentRecord();

try (RandomAccessFile raf = new RandomAccessFile(file, "rw")) {

raf.seek((recordNum - 1) \* RAFStudentRecord.SIZE);

record.read(raf);

record.setFirstName(firstName);

record.setLastName(lastName);

record.setGpa(gpaValue);

raf.seek((recordNum - 1) \* RAFStudentRecord.SIZE);

record.write(raf);

raf.close();

} catch (EOFException eof) {

System.out.println("End of File Reached");

} catch (IOException io) {

System.out.println(io.getMessage());

}

}

}

Comment out the call to the updateRecord method in the test class.

1. Create a class DeleteRAFRecord and add in the following code:

package solution;

import java.io.\*;

public class DeleteRAFRecord {

public static void deleteRecord(File file, int recordNum) {

RAFStudentRecord record = new RAFStudentRecord();

try {

// seek moves the file position pointer for object output

// based on calculation below

// Student numbers entered should be between 1 and 100

RandomAccessFile raf = new RandomAccessFile(file, "rw");

raf.seek((recordNum - 1) \* RAFStudentRecord.SIZE);

record.read(raf);

// check that there is data in the record

if (record.getSnum() == 0) {

System.out.println("ERROR: That record does not exist \n");

return;

}

raf.seek((recordNum - 1) \* RAFStudentRecord.SIZE);

record.setSnum(0); // reset the student number

record.write(raf); // update the record

System.out.println("Record Deleted");

raf.close();

} catch (EOFException eof) {

System.out.println(eof.getMessage());

} catch (IOException io) {

System.out.println(io.getMessage());

}

}

}

Comment out the call to the deleteRecord method in the test class.

Run the program and test if the methods work.