Create a folder **EServer** just for the files of this project

Globals.java
create Globals class (non-instantiating)

include all the following:

declare global string constant MESSAGES_FILE = "_messages.txt";
declare global RandomAccessFile msg variable for the messages file, initialized to null

declare global int constant PROCESS_OK with a value of 0
declare global int constant PROCESS_ERROR with a value of -1

declare global int variable totalRecordsInMessageFile that keeps the total records in the message file, initialized to -1

declare global constant RECORD_DATA_LEN with a value of 80
declare global constant NEXT_RECORD_LEN with a value of 4
declare global constant RECORD_LEN with a value of RECORD_DATA_LEN + NEXT_RECORD_LEN (be sure to use the constant names, not actual numbers)

FileIO.java
FileIO class (non-instantiating class)

methods

```
public static int openMessagesFile(String fileName)
     open file with global name Globals.msg and read/write attributes
     set Globals.totalRecordsInMessageFile to
                         (int) (length of messages file / length of one record)
          (the casting is needed because the length of a file is returned as a long)

     if all goes well return Globals.PROCESS_OK
     else return Globals.PROCESS_ERROR



public static int closeMessagesFile()
     close Global.msg file

     if all goes well return Globals.PROCESS_OK
     else return Globals.PROCESS_ERROR
```

EmailServer.java

```
main() method {
     open/create the messages file using the method of the class FileIO

     if all goes well {
          write a few fixed sized records (about 5 of some RECORD_LEN bytes each)
          print the value of Globals.totalRecordsInMessageFile

          if errors happen during writing
               print out an error message

          close the messages file using the method of the class FileIO
     }
     else
          print an error saying that the file could not be opened
}
```

After running the program, check that there is a file "_messages.txt" with a size that is an exact multiple of the size of records you chose while writing. For example, if your record size is 20 and you wrote 4 records, the file should be exactly 80 bytes.

If you run the program again, be sure to manually delete the file first.