

## Structure of a message part II: Message class

	<b>data</b> (RECORD_DATA_LEN bytes)	<b>next</b> (pointer to next record: 4 byte int)
0	SJBOND0007SUZIECUTE000000000+Hello@Dear Susan, how are you today? I will try to w	3
1	D123456789ABCDEFGHI000000000+No more money@Today is the day I run out of money	-1
2	hrough this code :(	3
3	rite more frequently about my travels in the far east.	-1
4	SFAROUT345TURTLENAP000000000+Hi@Hi Turtle: I don't know how we are going to get t	2

The messages file has been updated in the Pickup folder and contains the table above.  
It's under the name

Course Notes Unit 3 2 Data Structures I Eserver Pseudo Code 11 \_messages.dat

We now modify the **Message** class to include all the fields we have defined in Globals.

### **private variables**

define and initialize the following; character variables are initialized to null and String variables to null strings

```
command is char
sender, receiver, dateTime are String
marker is char
subject is String
eosMarker is char
text is String
```

### **constructors**

the default constructor contains a reinitialization of all the private variables

the following constructor is also needed:

```
public Message(String s) {
    make a call to setMessage(s)
}
```

### **access methods**

we will create these methods on a as-needed basis; we do not need to create all access methods right now. We need the following:

```
// minimize the use of number constants; use symbolic constants as much as possible
public void setMessage(String s) {
    command = character at COMMAND_POS position of s
    sender  = substring from SENDER_POS and a length of SENDER_LEN
    receiver = substring from RECEIVER_POS with a length of RECEIVER_LEN
```

```
    dateTime = ...
    marker    = ...
    subject   = substring from FIRST_RECORD_MARKER_POS + 1 up to but excluding
                                                    END_OF_SUBJECT_MARKER
    eosMarker = ...
    text      = substring from END_OF_SUBJECT_MARKER to end of s
}

public String getMessage() {
    return a string of all the fields in the correct order
}

public char getCommand() {
    return command
}

public String getSender() {
    return sender
}

public String getReceiver() {
    return receiver
}

public String getDateTime() {
    return dateTime
}
```

```
public String getId() {  
    return sender + receiver + dateTime  
}
```

**change to the method readFromMessagesFile() (Message class)**

the last line of the method is

```
text = data
```

it needs to be changed to

```
setMessage(data)
```

**toString() method**

return a string with every field on a separate line, with a label and a nice readable format. This is crucial for the testing that is coming up. Be sure to have a readable output, something like this:

```
Command      : S  
Sender       : 766564556  
Receiver     : Captain77  
Date/Time    : Sat 3 Feb 2017 7:23am ETD  
Marker       : +  
Subject      : I'm coming to the rescue  
EOS Marker   : @  
Message Text : Do not worry about anything. When I'm there all is well. When I am not,  
              no one cares. So I think I will go to sleep.
```



## EmailServer

For all tests be sure to:

- 1) Use the file `_messages.dat` in the `PickUp` folder. This file contains the table above (page 1)
- 2) Use the following template in the main program

```
main() {  
    declare and instantiate an object of the Message class, for example message  
    instantiate the availableList to an empty list  
  
    open the messages file (use FileIO.openMessagesFile() method; do not open directly)  
  
    if successful  
        < ...  
        write code here using the different tests in the table below  
        ... >  
  
        close the messages file  
    end if  
}
```

- 3) check the expected results as outlined in the table below

**Tests**

test	conditions	code being tested	expected results
1	read message in record 0 print the message object	setMessage() method	see output in row below
Command : S Sender : JBOND0007 Receiver : SUZIECUTE Date/Time : 00000000 Marker : + Subject : Hello EOS Marker : @ Message text: Dear Susan, how are you today? I will try to write more frequently about my travels in the far east.			
2	read message in record 5  for all these methods we place a call and print the returned result  getMessage() getCommand() getSender() getReceiver() getDateTime() getId()	All these methods  getMessage() getCommand() getSender() getReceiver() getDateTime() getId()	The correct field needs to output for all of these methods