The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming I
Assignment 1 Grade Sheet
2020/2021 Semester 2

JeanPaul Lezama (816017887)
42.09%
F2

Criteria	Mark
VirtualMeetingSystem	18.0
VirtualRoom	16.0
BreakoutRoom	17.0
Participant	6.0
Bonuses	+5.0
Deductions	-25.8
Total (out of 86.0)	**36.2**

Grade Changes

Observe any deductions or bonuses that you have incurred or earned.

Bonuses	
Question Bonus: allocateParticipants(String)	+5.0

Deductions	
Plagiarism Detected: "Cheater - First Infraction" from cluster with 3 other persons (-30%)	-25.8

VirtualMeetingSystem Class

Passed 8/10; Partially passed 1/10; Failed 1/10.

Method	**`createVirtualRoom(String)`**	2.0 / 2.0
Method	**`addParticipant(String, int)`**	2.0 / 2.0
Method	**`listParticipants(int)`**	3.0 / 3.0
Method	**`openBreakoutRoom(int)`**	2.0 / 2.0
Method	**`closeBreakoutRoom(int)`**	2.0 / 2.0
Method	**`listAllBreakoutRooms()`**	2.0 / 2.0
Method	**`findParticipantBreakoutRoom(String)`**	2.0 / 2.0
Method	**`listParticipantsInAllBreakoutRooms()`**	2.0 / 2.0
Method	**`allocateParticipants(String)`**	0.0 / 10.0

Does not work as anticipated: produced java.lang.NullPointerException: Cannot invoke "String.contains(java.lang.CharSequence)" because "participantID" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check `VirtualMeetingSystem()` constructor exists and is accessible ✓

Check `loadParticipantData(String)` method exists and is accessible	✓
Check `createVirtualRoom(String)` method exists and is accessible	✓
Check `public void allocateParticipants(String)` method exists and defined properly	✓
Check `VirtualMeetingSystem()` constructor creates instances	✓
Check `loadParticipantData(String)` method runs with args `("src/a1/test/resources/participant.dat")`	✓
Check `createVirtualRoom(String)` method runs with args `("VirtualRoom")`	✓
Check `allocateParticipants(String)` method runs with args `("RR")`	✓

Method **`loadParticipantData(String)`** 1.0 / 5.0

Does not alter the fileInfo attribute as it should. We expect your method to change the values of a particular set of instance attributes, but yours doesn't, or it does it in an unanticipated way.

Check `VirtualMeetingSystem()` constructor exists and is accessible	✓
Check `public void loadParticipantData(String)` method exists and defined properly	✓
Check `VirtualMeetingSystem()` constructor creates instances	✓
Check `loadParticipantData(String)` method runs with args `("src/a1/test/resources/participant.dat")`	✓
Check first attribute with type `String[]` equals `an array with size 50`	✓
Check `fileInfo` attribute equals `an array with size 50` +1.0	✓
Check first attribute with type `String[]` equals `not an array containing null` +1.0	✓
Check `fileInfo` attribute equals `not an array containing null`	✗

VirtualRoom Class

Passed 8/14; Partially passed 2/14; Failed 4/14.

Attribute **`breakoutRooms`** 1.0 / 1.0

Attribute **`name`** 1.0 / 1.0

Method	**listBreakoutRooms()**	2.0 / 2.0
Method	**findBreakoutRoom(int)**	2.0 / 2.0
Method	**createBreakoutRooms()**	2.0 / 2.0
Method	**findParticipantBreakoutRoom(String)**	2.0 / 2.0
Method	**listParticipantsInBreakoutRoom(int)**	2.0 / 2.0
Method	**addParticipantToBreakoutRoom(String, int)**	2.0 / 2.0

Attribute	**breakoutRoomLimit**	0.0 / 1.0

Does not have final modifier. We expect your attribute or method to have the final keyword indicating that it is a constant attribute or an un-overridable method, but yours doesn't.

Check `private final int breakoutRoomLimit` attribute exists and defined properly ✗

Constructor	**VirtualRoom(String, int)**	1.0 / 2.0

Does not initialize the breakoutRoomLimit attribute. We expect your constructor to initialize a particular set of instance attributes, but yours doesn't.

Check `public VirtualRoom(String, int)` constructor exists and defined properly ✓

Check `VirtualRoom(String, int)` constructor creates instances with args `("VirtualRoom", 10)` +0.5 ✓

Check `name` attribute equals `"VirtualRoom"` +0.5 ✓

Check `breakoutRoomLimit` attribute equals `10` ✗

Constructor	**VirtualRoom(String)**	1.0 / 2.0

Does not initialize the breakoutRoomLimit attribute. We expect your constructor to initialize a particular set of instance attributes, but yours doesn't.

Check `public VirtualRoom(String)` constructor exists and defined properly ✓

Check `VirtualRoom(String)` constructor creates instances with args `("VirtualRoom")` +0.5 ✓

Check `name` attribute equals `"VirtualRoom"` +0.5 ✓

Check `breakoutRoomLimit` attribute equals `5` ✗

Method **`openBreakoutRoom(int)`** 0.0 / 2.0

Cannot be further tested because testing relies on another method that Does not alter the breakoutRooms attribute as it should. We expect your method to change the values of a particular set of instance attributes, but yours doesn't, or it does it in an unanticipated way.

Check `VirtualRoom(String)` constructor exists and is accessible ✓

Check `createBreakoutRooms()` method exists and is accessible ✓

Check `closeBreakoutRoom(int)` method exists and is accessible ✓

Check `breakoutRooms` attribute exists ✓

Check `public boolean openBreakoutRoom(int)` method exists and defined properly ✓

Check `VirtualRoom(String)` constructor creates instances with args `("VirtualRoom")` ✓

Check `createBreakoutRooms()` method runs ✓

Check `closeBreakoutRoom(int)` method runs with args `(1)` ✓

Check `openBreakoutRoom(int)` method runs with args `(1)` ✓

Check `breakoutRooms` attribute equals `an array containing hasProperty("open", true)` ✗

Method **`closeBreakoutRoom(int)`** 0.0 / 2.0

Cannot be further tested because testing relies on another method that Does not alter the breakoutRooms attribute as it should. We expect your method to change the values of a particular set of instance attributes, but yours doesn't, or it does it in an unanticipated way.

Check `VirtualRoom(String)` constructor exists and is accessible ✓

Check `createBreakoutRooms()` method exists and is accessible ✓

Check `breakoutRooms` attribute exists ✓

Check `public boolean closeBreakoutRoom(int)` method exists and defined properly ✓

Check `VirtualRoom(String)` constructor creates instances with args `("VirtualRoom")` ✓

Check `createBreakoutRooms()` method runs ✓

Check `closeBreakoutRoom(int)` method runs with args `(1)` ✓

Check `breakoutRooms` attribute equals `an array containing hasProperty("open", false)` ✗

Method	**getNumberOfBreakoutRooms()**	0.0 / 1.0

Is not testable because testing relies on the breakoutRoomLimit attribute that is not an instance attribute. We expect your attribute to be defined without the static keyword indicating that it is an instance attribute, but yours isn't.

Check `VirtualRoom(String)` constructor exists and is accessible ✓

Check `breakoutRoomLimit` attribute exists ✗

BreakoutRoom Class

Passed 11/16; Partially passed 1/16; Failed 4/16.

Attribute	**breakoutRoomID**	1.0 / 1.0
Attribute	**participants**	1.0 / 1.0
Attribute	**numberOfParticipants**	1.0 / 1.0
Attribute	**open**	1.0 / 1.0
Attribute	**breakoutRoomNumberCounter**	1.0 / 1.0
Constructor	**BreakoutRoom(String)**	3.0 / 3.0
Method	**findParticipant(String)**	2.0 / 2.0
Method	**toString()**	2.0 / 2.0
Method	**addParticipant(String)**	2.0 / 2.0
Method	**openBreakoutRoom()**	1.0 / 1.0
Method	**closeBreakoutRoom()**	1.0 / 1.0
Attribute	**breakoutRoomSize**	0.0 / 1.0

Does not have final modifier. We expect your attribute or method to have the final keyword indicating that it is a constant attribute or an un-overridable method, but yours doesn't.

Check `private final int breakoutRoomSize` attribute exists and defined properly ✗

Method **`getBreakoutRoomID()`** 0.0 / 1.0

Is not defined. We expect your attribute, method or constructor to be defined in a particular way, but yours isn't or not defined at all.

Check `BreakoutRoom(String)` constructor exists and is accessible ✓

Check `breakoutRoomID` attribute exists ✓

Check `public String getBreakoutRoomID()` method exists and defined properly ✗

Method **`getOpen()`** 0.0 / 1.0

Is not defined. We expect your attribute, method or constructor to be defined in a particular way, but yours isn't or not defined at all.

Check `BreakoutRoom(String)` constructor exists and is accessible ✓

Check `open` attribute exists ✓

Check `public boolean getOpen()` method exists and defined properly ✗

Method **`getNumberOfParticipants()`** 0.0 / 1.0

Is not defined. We expect your attribute, method or constructor to be defined in a particular way, but yours isn't or not defined at all.

Check `BreakoutRoom(String)` constructor exists and is accessible ✓

Check `numberOfParticipants` attribute exists ✓

Check `public int getNumberOfParticipants()` method exists and defined properly ✗

Method **`listParticipants()`** 1.0 / 2.0

Returns correct information in an incorrect format. We expect your method to return a particular value in a given format, but instead yours returns the value in another format.

Check `BreakoutRoom(String)` constructor exists and is accessible ✓

Check `addParticipant(String)` method exists and is accessible	✓
Check `openBreakoutRoom()` method exists and is accessible	✓
Check `breakoutRoomID` attribute exists	✓
Check `public String listParticipants()` method exists and defined properly	✓
Check `BreakoutRoom(String)` constructor creates instances with args `("Room1")`	✓
Check `openBreakoutRoom()` method runs	✓
Check `addParticipant(String)` method runs with args `("10000000")`	✓
Check `listParticipants()` method returns string containing attribute `breakoutRoomID` +1.0	✓
Check `listParticipants()` method returns string containing attribute `participants` in format `"participant_1.toString() \n participant_2.toString() \n ... participant_n.toString()"`	✗

Participant Class

Passed 4/5; Failed 1/5.

Attribute	**`participantID`**	1.0 / 1.0
Constructor	**`Participant(String)`**	2.0 / 2.0
Method	**`toString()`**	2.0 / 2.0
Method	**`getParticipantID()`**	1.0 / 1.0
Method	**`verifyID(String)`**	0.0 / 2.0

Is not a class method. We expect your method to be defined with the static keyword indicating that it is an class method, but yours isn't.

Check `public static boolean verifyID(String)` method exists and defined properly	✗