



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

Shalana Seuraj (816021684)

100.00%

A+

Criteria	Mark
VirtualMeetingSystem	28.0
VirtualRoom	23.5
BreakoutRoom	21.0
Participant	8.0
Bonuses	+10.0
Total (out of 86.0)	86.0

Grade Changes

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

Bonuses	
Merit Bonus: Early submission	+5.0
Question Bonus: allocateParticipants(String)	+5.0

VirtualMeetingSystem Class

Passed 9/10; Partially passed 1/10.

Method	<code>createVirtualRoom(String)</code>	2.0 / 2.0
Method	<code>allocateParticipants(String)</code>	10.0 / 10.0
Method	<code>addParticipant(String, int)</code>	2.0 / 2.0
Method	<code>listParticipants(int)</code>	3.0 / 3.0
Method	<code>openBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>closeBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>listAllBreakoutRooms()</code>	2.0 / 2.0
Method	<code>findParticipantBreakoutRoom(String)</code>	2.0 / 2.0
Method	<code>listParticipantsInAllBreakoutRooms()</code>	2.0 / 2.0
Method	<code>loadParticipantData(String)</code>	1.0 / 5.0
Does not alter the participantsIDs 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 <code>VirtualMeetingSystem()</code> constructor exists and is accessible		✓
Check <code>public void loadParticipantData(String)</code> method exists and defined properly		✓

Check <code>VirtualMeetingSystem()</code> constructor creates instances	✓
Check <code>loadParticipantData(String)</code> method runs with args (<code>"src/al/test/resources/participant.dat"</code>)	✓
Check first attribute with type <code>String[]</code> equals an array with size 50	✓
Check <code>participantsIDs</code> attribute equals an array with size 50 +1.0	✓
Check first attribute with type <code>String[]</code> equals not an array containing null +1.0	✓
Check <code>participantsIDs</code> attribute equals not an array containing null	✗

VirtualRoom Class

Passed 13/14; Partially passed 1/14.

Attribute	<code>breakoutRooms</code>	1.0 / 1.0
Attribute	<code>breakoutRoomLimit</code>	1.0 / 1.0
Attribute	<code>name</code>	1.0 / 1.0
Constructor	<code>VirtualRoom(String, int)</code>	2.0 / 2.0
Constructor	<code>VirtualRoom(String)</code>	2.0 / 2.0
Method	<code>findBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>createBreakoutRooms()</code>	2.0 / 2.0
Method	<code>openBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>closeBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>findParticipantBreakoutRoom(String)</code>	2.0 / 2.0
Method	<code>getNumberOfBreakoutRooms()</code>	1.0 / 1.0
Method	<code>listParticipantsInBreakoutRoom(int)</code>	2.0 / 2.0

Method	<code>addParticipantToBreakoutRoom(String, int)</code>	2.0 / 2.0
Method	<code>listBreakoutRooms()</code>	1.5 / 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 <code>VirtualRoom(String)</code> constructor exists and is accessible		✓
Check <code>createBreakoutRooms()</code> method exists and is accessible		✓
Check <code>public String listBreakoutRooms()</code> method exists and defined properly		✓
Check <code>VirtualRoom(String)</code> constructor creates instances with args (<code>"VirtualRoom"</code>)		✓
Check <code>createBreakoutRooms()</code> method runs		✓
Check <code>listBreakoutRooms()</code> method returns string containing attribute <code>name</code> +1.5		✓
Check <code>listBreakoutRooms()</code> method returns string containing attribute <code>breakoutRooms</code> in format <code>"breakoutRoom_1.toString() \n breakoutRoom_2.toString() \n ... breakoutRoom_n.toString()"</code>		✗

BreakoutRoom Class

Passed 15/16; Partially passed 1/16.

Attribute	<code>breakoutRoomID</code>	1.0 / 1.0
Attribute	<code>breakoutRoomSize</code>	1.0 / 1.0
Attribute	<code>participants</code>	1.0 / 1.0
Attribute	<code>numberOfParticipants</code>	1.0 / 1.0
Attribute	<code>open</code>	1.0 / 1.0
Attribute	<code>breakoutRoomNumberCounter</code>	1.0 / 1.0
Constructor	<code>BreakoutRoom(String)</code>	3.0 / 3.0
Method	<code>findParticipant(String)</code>	2.0 / 2.0

Method	<code>toString()</code>	2.0 / 2.0
Method	<code>getBreakoutRoomID()</code>	1.0 / 1.0
Method	<code>getOpen()</code>	1.0 / 1.0
Method	<code>addParticipant(String)</code>	2.0 / 2.0
Method	<code>openBreakoutRoom()</code>	1.0 / 1.0
Method	<code>closeBreakoutRoom()</code>	1.0 / 1.0
Method	<code>getNumberOfParticipants()</code>	1.0 / 1.0
Method	<code>listParticipants()</code>	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 <code>BreakoutRoom(String)</code> constructor exists and is accessible		✓
Check <code>addParticipant(String)</code> method exists and is accessible		✓
Check <code>openBreakoutRoom()</code> method exists and is accessible		✓
Check <code>breakoutRoomID</code> attribute exists		✓
Check <code>public String listParticipants()</code> method exists and defined properly		✓
Check <code>BreakoutRoom(String)</code> constructor creates instances with args ("Room1")		✓
Check <code>openBreakoutRoom()</code> method runs		✓
Check <code>addParticipant(String)</code> method runs with args ("10000000")		✓
Check <code>listParticipants()</code> method returns string containing attribute <code>breakoutRoomID + 1.0</code>		✓
Check <code>listParticipants()</code> method returns string containing attribute <code>participants</code> in format "participant_1.toString() \n participant_2.toString() \n ... participant_ n.toString() "		✗

Participant Class

Passed 5/5.

Attribute	<code>participantID</code>	1.0 / 1.0
Constructor	<code>Participant(String)</code>	2.0 / 2.0
Method	<code>toString()</code>	2.0 / 2.0
Method	<code>verifyID(String)</code>	2.0 / 2.0
Method	<code>getParticipantID()</code>	1.0 / 1.0