



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

Kadeem Compton (814002641)

96.51%

A+

Criteria	Mark
VirtualMeetingSystem	27.0
VirtualRoom	23.0
BreakoutRoom	21.5
Participant	7.5
Bonuses	+5.0
Deductions	-1.0
Total (out of 86.0)	83.0

Grade Changes

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

Bonuses	
Question Bonus: allocateParticipants(String)	+5.0
Deductions	
Formatting Error: Incorrect archive format (please use .zip)	-1

VirtualMeetingSystem Class

Passed 9/10; Failed 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>	0.0 / 5.0
Does not alter the participantsArray 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 <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/a1/test/resources/participant.dat"</code>)	✓
Check first attribute with type <code>String[]</code> equals an array with size 50	✓
Check <code>participantsArray</code> attribute equals an array with size 50	✗

VirtualRoom Class

Passed 13/14; Failed 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>listBreakoutRooms()</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>listParticipantsInBreakoutRoom(int)</code>	2.0 / 2.0
Method	<code>addParticipantToBreakoutRoom(String, int)</code>	2.0 / 2.0

Method	<code>getNumberOfBreakoutRooms()</code>	0.0 / 1.0
Returns abnormal values. We expect your method to return a particular value, but instead yours returns a value that is not feasible nor valid.		
Check <code>VirtualRoom(String)</code> constructor exists and is accessible		✓
Check <code>breakoutRoomLimit</code> attribute exists		✓
Check <code>public int getNumberOfBreakoutRooms()</code> method exists and defined properly		✓
Check <code>VirtualRoom(String)</code> constructor creates instances with args (<code>"VirtualRoom"</code>)		✓
Check <code>getNumberOfBreakoutRooms()</code> method returns value equal to attribute <code>breakoutRoomLimit</code>		✗

BreakoutRoom Class

Passed 14/16; Partially passed 2/16.

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
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
Constructor	<code>BreakoutRoom(String)</code>	3.5 / 3.0
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 <code>"participant_1.toString() \n participant_2.toString() \n ... participant_</code> <code>n.toString() "</code>		✗

Participant Class

Passed 4/5; Partially passed 1/5.

Attribute	<code>participantID</code>	1.0 / 1.0
Constructor	<code>Participant(String)</code>	2.0 / 2.0

Method	<code>verifyID(String)</code>	2.0 / 2.0
Method	<code>getParticipantID()</code>	1.0 / 1.0
Method	<code>toString()</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>Participant(String)</code> constructor exists and is accessible		✓
Check <code>participantID</code> attribute exists		✓
Check <code>public String toString()</code> method exists and defined properly		✓
Check <code>Participant(String)</code> constructor creates instances with args ("12345678")		✓
Check <code>toString()</code> method returns string containing attribute <code>participantID</code> +1.5		✓
Check <code>toString()</code> method returns string containing attribute <code>participantID</code> in format <code>Participant: participantID</code>		✗