



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

Christine Ramdhanie (816018588)

42.44%

**F2**

Criteria	Mark
VirtualMeetingSystem	2.0
VirtualRoom	6.0
BreakoutRoom	20.5
Participant	8.0
<b>Total (out of 86.0)</b>	<b>36.5</b>

## VirtualMeetingSystem Class

Passed 1/10; Failed 9/10.

Method	<code>createVirtualRoom(String)</code>	2.0 / 2.0
--------	--	-----------

Method	<code>loadParticipantData(String)</code>	0.0 / 5.0
--------	--	-----------

Does not work as anticipated: produced java.io.FileNotFoundException: participant.dat (No such file or directory). We expect your method to run without problems, but instead yours contains bad code that creates problems.

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> )	✗
--	---

Method	<code>allocateParticipants(String)</code>	0.0 / 10.0
--------	---	------------

Cannot be further tested because testing relies on the `loadParticipantData` method that does not work as anticipated: produced java.io.FileNotFoundException: participant.dat (No such file or directory). We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓
--	---

Check <code>loadParticipantData(String)</code> method exists and is accessible	✓
--	---

Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓
--	---

Check <code>public void allocateParticipants(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> )	✗
--	---

Method	<code>addParticipant(String, int)</code>	0.0 / 2.0
--------	--	-----------

Cannot be further tested because testing relies on the `loadParticipantData` method that does not work as anticipated: produced java.io.FileNotFoundException: participant.dat (No such file or directory). We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓
Check <code>loadParticipantData(String)</code> method exists and is accessible	✓
Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓
Check <code>openBreakoutRoom(int)</code> method exists and is accessible	✓
Check <code>public boolean addParticipant(String, int)</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> )	✗

Method	<code>listParticipants(int)</code>	0.0 / 3.0
Cannot be further tested because testing relies on the <code>openBreakoutRoom</code> method that does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null</code> . We expect your method to run without problems, but instead yours contains bad code that creates problems.		
Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓	
Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓	
Check <code>openBreakoutRoom(int)</code> method exists and is accessible	✓	
Check <code>addParticipant(String, int)</code> method exists and is accessible	✓	
Check <code>public String listParticipants(int)</code> method exists and defined properly	✓	
Check <code>VirtualMeetingSystem()</code> constructor creates instances	✓	
Check <code>createVirtualRoom(String)</code> method runs with args ( <code>"VirtualRoom"</code> )	✓	
Check <code>openBreakoutRoom(int)</code> method runs with args ( <code>1</code> )	✗	

Method	<code>openBreakoutRoom(int)</code>	0.0 / 2.0
Does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null</code> . We expect your method to run without problems, but instead yours contains bad code that creates problems.		
Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓	
Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓	

Check <code>public boolean openBreakoutRoom(int)</code> method exists and defined properly	✓
Check <code>VirtualMeetingSystem()</code> constructor creates instances	✓
Check <code>createVirtualRoom(String)</code> method runs with args ( "VirtualRoom" )	✓
Check <code>openBreakoutRoom(int)</code> method runs with args ( 1 )	✗

Method	<code>closeBreakoutRoom(int)</code>	0.0 / 2.0
--------	-------------------------------------	-----------

Does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓
Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓
Check <code>public boolean closeBreakoutRoom(int)</code> method exists and defined properly	✓
Check <code>VirtualMeetingSystem()</code> constructor creates instances	✓
Check <code>createVirtualRoom(String)</code> method runs with args ( "VirtualRoom" )	✓
Check <code>closeBreakoutRoom(int)</code> method runs with args ( 1 )	✗

Method	<code>listAllBreakoutRooms()</code>	0.0 / 2.0
--------	-------------------------------------	-----------

Does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check <code>VirtualMeetingSystem()</code> constructor exists and is accessible	✓
Check <code>createVirtualRoom(String)</code> method exists and is accessible	✓
Check <code>public String listAllBreakoutRooms()</code> method exists and defined properly	✓
Check <code>VirtualMeetingSystem()</code> constructor creates instances	✓
Check <code>createVirtualRoom(String)</code> method runs with args ( "VirtualRoom" )	✓
Check <code>listAllBreakoutRooms()</code> method returns a string containing "VirtualRoom"	✗

Method	<code>findParticipantBreakoutRoom(String)</code>	0.0 / 2.0
--------	--	-----------

Cannot be further tested because testing relies on the openBreakoutRoom method that does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" 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 createVirtualRoom(String) method exists and is accessible	✓
Check openBreakoutRoom(int) method exists and is accessible	✓
Check addParticipant(String, int) method exists and is accessible	✓
Check public String findParticipantBreakoutRoom(String) method exists and defined properly	✓
Check VirtualMeetingSystem() constructor creates instances	✓
Check createVirtualRoom(String) method runs with args ("VirtualRoom")	✓
Check openBreakoutRoom(int) method runs with args (1)	✗

Method	<b>listParticipantsInAllBreakoutRooms()</b>	0.0 / 2.0
--------	---	-----------

Cannot be further tested because testing relies on the openBreakoutRoom method that does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" 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 createVirtualRoom(String) method exists and is accessible	✓
Check openBreakoutRoom(int) method exists and is accessible	✓
Check addParticipant(String, int) method exists and is accessible	✓
Check public String listParticipantsInAllBreakoutRooms() method exists and defined properly	✓
Check VirtualMeetingSystem() constructor creates instances	✓
Check createVirtualRoom(String) method runs with args ("VirtualRoom")	✓
Check openBreakoutRoom(int) method runs with args (1)	✗

## VirtualRoom Class

Passed 3/14; Partially passed 2/14; Failed 9/14.

Attribute	<code>breakoutRooms</code>	1.0 / 1.0
Attribute	<code>name</code>	1.0 / 1.0
Method	<code>createBreakoutRooms()</code>	2.0 / 2.0
Attribute	<code>breakoutRoomLimit</code>	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 <code>private final int breakoutRoomLimit</code> attribute exists and defined properly		✗
Constructor	<code>VirtualRoom(String, int)</code>	0.5 / 2.0
Does not initialize the name attribute. We expect your constructor to initialize a particular set of instance attributes, but yours doesn't.		
Check <code>public VirtualRoom(String, int)</code> constructor exists and defined properly		✓
Check <code>VirtualRoom(String, int)</code> constructor creates instances with args ( <code>"VirtualRoom"</code> , <code>10</code> ) +0.5		✓
Check <code>name</code> attribute equals <code>"VirtualRoom"</code>		✗
Constructor	<code>VirtualRoom(String)</code>	1.5 / 2.0
Does not initialize the breakoutRooms attribute. We expect your constructor to initialize a particular set of instance attributes, but yours doesn't.		
Check <code>public VirtualRoom(String)</code> constructor exists and defined properly		✓
Check <code>VirtualRoom(String)</code> constructor creates instances with args ( <code>"VirtualRoom"</code> ) +0.5		✓
Check <code>name</code> attribute equals <code>"VirtualRoom"</code> +0.5		✓
Check <code>breakoutRoomLimit</code> attribute equals <code>5</code> +0.5		✓

Check breakoutRooms attribute equals an array with size 5

✗

Method

listBreakoutRooms()

0.0 / 2.0

Does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check VirtualRoom(String) constructor exists and is accessible

✓

Check createBreakoutRooms() method exists and is accessible

✓

Check public String listBreakoutRooms() method exists and defined properly

✓

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

✓

Check createBreakoutRooms() method runs

✓

Check listBreakoutRooms() method returns string containing attribute name

✗

Method

findBreakoutRoom(int)

0.0 / 2.0

Does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check VirtualRoom(String) constructor exists and is accessible

✓

Check createBreakoutRooms() method exists and is accessible

✓

Check public BreakoutRoom findBreakoutRoom(int) method exists and defined properly

✓

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

✓

Check createBreakoutRooms() method runs

✓

Check findBreakoutRoom(int) method returns not null with args (1)

✗

Method

openBreakoutRoom(int)

0.0 / 2.0

Cannot be further tested because testing relies on the closeBreakoutRoom method that does not work as anticipated: produced java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null. We expect your method to run without problems, but instead yours contains bad code that creates problems.

Check VirtualRoom(String) constructor exists and is accessible

✓

Check <code>createBreakoutRooms()</code> method exists and is accessible	✓
Check <code>closeBreakoutRoom(int)</code> method exists and is accessible	✓
Check <code>breakoutRooms</code> attribute exists	✓
Check <code>public boolean openBreakoutRoom(int)</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>closeBreakoutRoom(int)</code> method runs with args ( <code>1</code> )	✗

Method	<code>closeBreakoutRoom(int)</code>	0.0 / 2.0
Does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null.</code> We expect your method to run without problems, but instead yours contains bad code that creates problems.		
Check <code>VirtualRoom(String)</code> constructor exists and is accessible	✓	
Check <code>createBreakoutRooms()</code> method exists and is accessible	✓	
Check <code>breakoutRooms</code> attribute exists	✓	
Check <code>public boolean closeBreakoutRoom(int)</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>closeBreakoutRoom(int)</code> method runs with args ( <code>1</code> )	✗	

Method	<code>findParticipantBreakoutRoom(String)</code>	0.0 / 2.0
Cannot be further tested because testing relies on the <code>openBreakoutRoom</code> method that does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null.</code> We expect your method to run without problems, but instead yours contains bad code that creates problems.		
Check <code>VirtualRoom(String)</code> constructor exists and is accessible	✓	
Check <code>createBreakoutRooms()</code> method exists and is accessible	✓	
Check <code>listParticipantsInBreakoutRoom(int)</code> method exists and is accessible	✓	



Check <code>addParticipantToBreakoutRoom(String, int)</code> method exists and is accessible	✓
Check <code>public String findParticipantBreakoutRoom(String)</code> method exists and defined properly	✓
Check <code>VirtualRoom(String)</code> constructor creates instances with args ( "VirtualRoom" )	✓
Check <code>createBreakoutRooms()</code> method runs	✓
Check <code>openBreakoutRoom(int)</code> method runs with args ( 1 )	✗

Method	<code>getNumberOfBreakoutRooms()</code>	0.0 / 1.0
Does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null.</code> We expect your method to run without problems, but instead yours contains bad code that creates problems.		
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 ( "VirtualRoom" )		✓
Check <code>getNumberOfBreakoutRooms()</code> method returns value equal to attribute <code>breakoutRoomLimit</code>		✗

Method	<code>listParticipantsInBreakoutRoom(int)</code>	0.0 / 2.0
Does not work as anticipated: produced <code>java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null.</code> We expect your method to run without problems, but instead yours contains bad code that creates problems.		
Check <code>VirtualRoom(String)</code> constructor exists and is accessible		✓
Check <code>createBreakoutRooms()</code> method exists and is accessible		✓
Check <code>addParticipantToBreakoutRoom(String, int)</code> method exists and is accessible		✓
Check <code>public String listParticipantsInBreakoutRoom(int)</code> method exists and defined properly		✓
Check <code>VirtualRoom(String)</code> constructor creates instances with args ( "VirtualRoom" )		✓
Check <code>createBreakoutRooms()</code> method runs		✓

Check `listParticipantsInBreakoutRoom(int)` method returns not null with args (1)

✗

Method

`addParticipantToBreakoutRoom(String, int)`

0.0 / 2.0

Cannot be further tested because testing relies on the `openBreakoutRoom` method that does not work as anticipated: produced `java.lang.NullPointerException: Cannot load from object array because "this.breakoutRooms" is null`. We expect your method to run without problems, but instead yours contains bad code that creates problems.

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

✓

Check `createBreakoutRooms()` method exists and is accessible

✓

Check `listParticipantsInBreakoutRoom(int)` method exists and is accessible

✓

Check `public boolean addParticipantToBreakoutRoom(String, int)` method exists and defined properly

✓

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

✓

Check `createBreakoutRooms()` method runs

✓

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

✗

## BreakoutRoom Class

Passed 13/16; Partially passed 3/16.

Attribute

`breakoutRoomID`

1.0 / 1.0

Attribute

`breakoutRoomSize`

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

Method

`findParticipant(String)`

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>toString()</code>	1.0 / 2.0
<p>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.</p>		
Check <code>BreakoutRoom(String)</code> constructor exists and is accessible		✓
Check <code>breakoutRoomID</code> attribute exists		✓
Check <code>numberOfParticipants</code> attribute exists		✓
Check <code>public String toString()</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>toString()</code> method returns string containing attribute <code>breakoutRoomID +0.5</code>		✓
Check <code>toString()</code> method returns string containing attribute <code>numberOfParticipants +0.5</code>		✓
Check <code>toString()</code> method returns string containing attribute <code>breakoutRoomID</code> in format <code>breakoutRoomID OPEN</code>		✗
Method	<code>listParticipants()</code>	1.0 / 2.0
<p>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.</p>		
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