



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

Derwin Jack (816022295)

66.28%

**B**

Criteria	Mark
VirtualMeetingSystem	18.0
VirtualRoom	17.5
BreakoutRoom	11.0
Participant	5.5
Bonuses	+5.0
<b>Total (out of 86.0)</b>	<b>57.0</b>

## Grade Changes

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

Bonuses	
Question Bonus: allocateParticipants(String)	+5.0

## VirtualMeetingSystem Class

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

Method	<code>createVirtualRoom(String)</code>	2.0 / 2.0
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>allocateParticipants(String)</code>	0.0 / 10.0
Does not work as anticipated: produced java.lang.NullPointerException: Cannot invoke "String.length()" because "participantID" 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>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> )	✓
Check <code>createVirtualRoom(String)</code> method runs with args ( <code>"VirtualRoom"</code> )	✓
Check <code>allocateParticipants(String)</code> method runs with args ( <code>"RR"</code> )	✓

Method	<code>loadParticipantData(String)</code>	1.0 / 5.0
Does not alter the parr 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>parr</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>parr</code> attribute equals not an array containing null		✗

## VirtualRoom Class

Passed 9/14; Partially passed 1/14; Failed 4/14.

Attribute	<code>breakoutRooms</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>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
Attribute	<code>breakoutRoomLimit</code>	0.0 / 1.0
<p>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.</p> <p>Check <code>private final int breakoutRoomLimit</code> attribute exists and defined properly</p>		
Method	<code>openBreakoutRoom(int)</code>	0.0 / 2.0
<p>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.</p> <p>Check <code>VirtualRoom(String)</code> constructor exists and is accessible</p> <p>Check <code>createBreakoutRooms()</code> method exists and is accessible</p> <p>Check <code>closeBreakoutRoom(int)</code> method exists and is accessible</p> <p>Check <code>breakoutRooms</code> attribute exists</p> <p>Check <code>public boolean openBreakoutRoom(int)</code> method exists and defined properly</p> <p>Check <code>VirtualRoom(String)</code> constructor creates instances with args (<code>"VirtualRoom"</code>)</p> <p>Check <code>createBreakoutRooms()</code> method runs</p> <p>Check <code>closeBreakoutRoom(int)</code> method runs with args (<code>1</code>)</p> <p>Check <code>openBreakoutRoom(int)</code> method runs with args (<code>1</code>)</p> <p>Check <code>breakoutRooms</code> attribute equals an array containing <code>hasProperty("open", true)</code></p>		
Method	<code>closeBreakoutRoom(int)</code>	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
--------	----------------------------	-----------

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 VirtualRoom(String) constructor exists and is accessible	✓
Check breakoutRoomLimit attribute exists	✓
Check public int getNumberOfBreakoutRooms() method exists and defined properly	✓
Check VirtualRoom(String) constructor creates instances with args ("VirtualRoom")	✓
Check getNumberOfBreakoutRooms() method returns value equal to attribute breakoutRoomLimit	✗

Method	listBreakoutRooms()	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 VirtualRoom(String) constructor exists and is accessible	✓
Check createBreakoutRooms() method exists and is accessible	✓
Check public String listBreakoutRooms() 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 8/16; Partially passed 1/16; Failed 7/16.

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>addParticipant(String)</code>	2.0 / 2.0
Method	<code>openBreakoutRoom()</code>	1.0 / 1.0
Method	<code>closeBreakoutRoom()</code>	1.0 / 1.0
Attribute	<code>breakoutRoomID</code>	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 <code>private String breakoutRoomID</code> attribute exists and defined properly		✗
Attribute	<code>breakoutRoomSize</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 `private final int breakoutRoomSize` attribute exists and defined properly

✗

#### Constructor

#### `BreakoutRoom(String)`

1.0 / 3.0

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

Check `public BreakoutRoom(String)` constructor exists and defined properly

✓

Check `BreakoutRoom(String)` constructor creates instances with args ( "Room1" ) +1.0

✓

Check `breakoutRoomID` attribute equals `not null`

✗

#### Method

#### `toString()`

0.0 / 2.0

Is not testable because testing relies on the `breakoutRoomID` attribute that 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

✗

#### Method

#### `getBreakoutRoomID()`

0.0 / 1.0

Is not testable because testing relies on the `breakoutRoomID` attribute that 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

✗

#### 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	<code>listParticipants()</code>	0.0 / 2.0
Is not testable because testing relies on the <code>breakoutRoomID</code> attribute that 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 <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		✗

Method	<code>getNumberOfParticipants()</code>	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 <code>BreakoutRoom(String)</code> constructor exists and is accessible		✓
Check <code>numberOfParticipants</code> attribute exists		✓
Check <code>public int getNumberOfParticipants()</code> method exists and defined properly		✗

## Participant Class

Passed 3/5; Partially passed 1/5; Failed 1/5.

Attribute	<code>participantID</code>	1.0 / 1.0
Constructor	<code>Participant(String)</code>	2.0 / 2.0
Method	<code>getParticipantID()</code>	1.0 / 1.0
Method	<code>verifyID(String)</code>	0.0 / 2.0
Is not a class method. We expect your method to be defined with the <code>static</code> keyword indicating that it is an class method, but yours isn't.		
Check <code>public static boolean verifyID(String)</code> method exists and defined properly		✗
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 `Participant(String)` constructor exists and is accessible ✓

Check `participantID` attribute exists ✓

Check `public String toString()` method exists and defined properly ✓

Check `Participant(String)` constructor creates instances with args ( "12345678" ) ✓

Check `toString()` method returns string containing attribute `participantID +1.5` ✓

Check `toString()` method returns string containing attribute `participantID` in format  
`Participant: participantID` ✗