ID	Criterion	Marks	*Available*
1	Comments & annotations	10	*10*
3	Java conventions	19	*20*
4	Operation	32	*40*
5	OO design	30	*30*
6	Penalty: 3rd party	0	*-10*
7	Penalty: input	0	*-5*
9	Penalty: code printout	0	*-5*
	TOTAL	**91**	

General comments:
Well written and well-functioning code. However, as a result of putting all classes within the com.company package, and not maintaining proper directory structure as required by the use of this packge, simply executing javac BookingApp.java was not working.
Sequential IDs were always starting from 10 instead of 11. None of the lists were titled.
Code comments (/4): Code is commented clearly, but not excessively, and there are no points in code where you cannot easily tell what is going on.
Documentation comments (/4): All methods have complete JavaDoc (including @param, @returns @ throws etc) and all classes have JavaDoc at the top.
Annotations (/2): @Overrides' used for all methods implementing interface methods.
Naming (/10):
Attribute and variable names makes sense and are clear.
Method names make sense and are clear.
Class names make sense and are clear.
CamelCase used for classes (start uppercase).
CamelCase used for variables and methods (start lowercase), constants all uppercase CONSTANT_VALUE.
Structure (/10):
Indentation is consistent within Class files.
Indentation is consistent between Class files.
Methods are not short -- excessively long method bodies.
Code is not repeated (or there is very little repetition) -- private methods are called to reduce code duplication.
JavaDoc (html) generated at doc/ folder.
OO design (/30):
Attributes are all private (unless constants) -- i.e. well-encapsulated classes
A bookable room is associated with a room.
An assistant on shift is associated with an assistant.
University and Booking system are represented by distinct classes.
Booking is represented by its own class.
All type choices makes sense (e.g. integer types used to store integer numeric values, etc.).
Operation (/40) success:
Main menu screen follows the specification.
3 from menu takes you to remove bookable room.
Entering invalid ID does not remove bookable room, shows a meaningful msg, and allows entering new ID.
0 from remove bookable room screen returns to menu.
2 from menu takes you to add bookable room.
Entering invalid inputs does not add a bookable room, shows a meaningful msg, and allows entering new inputs.
Entering a valid inputs adds the bookable room, shows a meaningful msg, and allows entering new inputs.
0 from add bookable room screen returns to menu.
Entering a valid ID removes the bookable room, shows a meaningful msg, and allows entering new ID.
1 in the main menu takes you to the list of bookable rooms screen.
List bookable rooms shows at least 9 bookable rooms. There is at least one in each status: full, available and empty.
0 from list bookable room screen returns to menu.
6 from menu takes you to remove assistant on shift.
Entering invalid ID does not remove assistant on shift, shows a meaningful msg, and allows entering new ID.
0 from remove assistant on shift screen returns to menu.

5 from menu takes you to add assistant on shift.
Entering invalid inputs does not add an assistant on shift, shows a meaningful msg, and allows entering new inputs.
Entering a valid inputs adds the assistant on shift, shows a meaningful msg, and allows entering new inputs.
0 from add assistant on shift screen returns to menu.
Entering a valid ID removes the assistant on shift, shows a meaningful msg, and allows entering new ID.
3 in the main menu takes you to the list of assistants on shift screen.
List assistant on shift shows at least 6 assistants on shift. There is at least one busy and one free.
0 from list assistant on shift screen returns to menu.
9 from menu takes you to remove booking.
Entering invalid ID does not remove booking, shows a meaningful msg, and allows entering new ID.
0 from remove booking screen returns to menu.
8 from menu takes you to add booking.
Entering invalid inputs does not add a booking, shows a meaningful msg, and allows entering new inputs.
0 from add booking screen returns to menu.
10 from menu takes you to conclude booking.
Entering a valid ID concludes the booking, shows a meaningful msg, and allows entering new ID.
Entering a valid ID removes the booking, shows a meaningful msg, and allows entering new ID.
7 in the main menu takes you to the list of bookings screen.
List at least one booking scheduled and one completed.
0 from list booking screen returns to menu.
-1 finishes the application from any screen.

Operation FAIL:
The command javac BookingApp.java compiles successfully.
The command java BookingApp launches the booking system and show the main menu.
Remove bookable room screen follows the specification, including sequential IDs.
Add bookable room screen follows the specification, including sequential IDs.
List bookable rooms screen follows the specification.
Remove assistant on shift screen follows the specification, including sequential IDs.
Add assistant on shift screen follows the specification, including sequential IDs.
List assistants on shift screen follows the specification.
Remove booking screen follows the specification, including sequential IDs.
Add booking screen follows the specification, including sequential IDs.
Entering a valid inputs adds the booking, shows a meaningful msg, and allows entering new inputs.
Conclude booking screen follows the specification, including sequential IDs.
List bookings screen follows the specification.