System Operations and Operation Contracts

1. Operation: createOffering(space: Space, schedule: Schedule)

- **Cross-reference**: Use Case 1 - "Admin Makes Offerings Available"
- **Preconditions**:
  - The admin is logged into the system.
  - The space and schedule details are valid and provided.
- **Postconditions**:
  - If no conflicts exist, a new offering is created and added to the list of offerings.
  - If a conflict is found, the system returns a conflict message and no offering is created.

---

2. Operation: checkConflictingOffering(space: Space, schedule: Schedule)

- **Cross-reference**: Use Case 1 - "Admin Makes Offerings Available"
- **Preconditions**:
  - The space and schedule details are valid.
- **Postconditions**:
  - If a conflicting offering exists (i.e., another offering uses the same space and schedule), the system returns a conflict message.
  - If no conflict is found, the system allows the creation of the new offering.

---

3. Operation: addOffering(Offering)

- **Cross-reference**: Use Case 1 - "Admin Makes Offerings Available"
- **Preconditions**:
  - The system has validated that there are no conflicting offerings.
- **Postconditions**:
  - The offering is successfully added to the system and is available for selection by instructors.
  - The system sends a confirmation to the admin.

---

4. Operation: viewAvailableOfferings()

- **Cross-reference**: Use Case 2 - "Instructor Selects Lessons"
- **Preconditions**:
    - The instructor is logged into the system.
    - There are offerings that are not fully booked.
- **Postconditions**:
    - The system retrieves and displays a list of available offerings for the instructor to view.

---

5. Operation: selectOffering(Offering)

- **Cross-reference**: Use Case 2 - "Instructor Selects Lessons"
- **Preconditions**:
    - The instructor is logged into the system and is viewing the list of available offerings.
    - The offering is available (not already assigned to another instructor).
- **Postconditions**:
    - The selected offering is assigned to the instructor.
    - The system updates the list of available offerings to reflect the selection.
    - A confirmation is sent to the instructor.

---

6. Operation: setInstructor(Instructor)

- **Cross-reference**: Use Case 2 - "Instructor Selects Lessons"
- **Preconditions**:
    - The instructor has selected a valid offering.
- **Postconditions**:
    - The instructor is assigned to the offering.
    - The system updates the offering with the instructor's details and confirms the selection.

---

7. Operation: getAvailableOfferings()

- **Cross-reference**: Use Case 3 - "Public Views Offerings"
- **Preconditions**:
    - The system contains a list of offerings that are not fully booked.
- **Postconditions**:
    - The system retrieves and displays the list of available offerings for the public to view.

---

8. Operation: getOfferingsWithInstructors()

- **Cross-reference**: Use Case 3 - "Public Views Offerings"
- **Preconditions**:
    - The system contains offerings that have instructors assigned.
- **Postconditions**:
    - The system retrieves the list of offerings along with their assigned instructors and displays this to the public.