Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail

1. How can we communicate with other component in Angular?
2. What is @Input and @Output used for?
3. What is @ViewChild used for?
4. What is Local Variable in Angular?
5. What is DI in Angular and how it works?
6. What is Service in Angular and why do we need it?
7. What is Observable in Angular?
8. What is difference between Promise and Observable in Angular?
9. What are advantages of Observable in Angular?

Coding Questions:

Write code in Java to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

Submit screen shot in a doc or PDC for all assignment.

1. **Component Communication (don't use service for this assignment)**
   - Create three new components: GameControl, Odd and Even
   - The GameControl Component should have buttons to start and stop the game
   - When starting the game, an event (holding a incrementing number) should get emitted each second (ref = setInterval())
   - The event should be listenable from outside the component
   - When stopping the game, no more events should get emitted (clearInterval(ref))
   - A new Odd component should get created for every odd number emitted, the same should happen for the Even Component (on even numbers)
   - Simply output Odd - NUMBER or Even - NUMBER in the two components
   - Style the element (e.g. paragraph) holding your output text differently in both components
     - Set the text color of odd event to green
     - Set the text color of even event to red
   - Add bootstrap 3/4 to your application

Note: The output should look like
In 1st second:
Odd - 1
In 2nd second:
Odd - 1
Even - 2
In 3rd second:
Odd - 1
Odd - 3

Even - 2
In 4th second:
Odd - 1
Odd - 3
Even - 2

2. Service Practice

- Create two new components: ActiveUser, InactiveUser
- Create a service: UserService
- The UserService should have two list of string: activeUsers and inactiveUsers as initial values (At least 3 user in each list)
- All component communication should go through the UserService
- The ActiveUser component should display all activeUser in each line with a button to set the user as Inactive.
- If user clicks the button, the user should be removed from the active user list and add to inactive user list.
- Just like ActiveUser component, the InactiveUser component should have the exactly same functionality.
- Use BootStrap to add style to the list.

3. Make call to the restful endpoint in Spring application

- Create a Spring application which has a endpoint where we can get a list of accounts
  - You don't need to use database, just hard the value
  - Account should have three fields: accountNumber, accountName, isActive
- Create Angular Application to retrieve the account information from the Spring application
  - You should use Service to hold this logic, since it is not component related logic
  - Create a model/domain object to hold account information
  - Display the account information with following rule
    - Display all account information in one line
    - If the account is inactive, grey out the account information (set the background color)

Note: Please research the CORS (Cross-Origin Resource Sharing) before you start this assignment.
Here is a simple solution to this issue — use proxy in Angular.