```
/************ 11_EVENT_BUBBLING_DELEGATION ******/

/**
 * Event Bubbling:
 *  --> If a event is fired on some elment and that element fires the events of its all parents elements automatically, this nature is called Event Bubbling
 *
 * EXAMPLE:
 *  --> ROOT:
 *      -> LEVEL 1:
 *          -> LEVEL 2:
 *              -> LEVEL 3:
 *
 * --> if an event of LEVEL 3 is fired then the events on LEVEL 2 will be fired which causes the events of LEVEL 1 to be fired then finally ROOT LEVEL Events' will be fired. --> Due To Event Bubbling
 * --> Event Bubbling Can Be Stopped Using e.stopPropagation();
 */

/**
 * Card
 *  -> Card Content
 *      -> Card Title
 */
document.querySelector('.card-title')
    .addEventListener('click', function(e) {
        e.stopPropagation();
        console.log('card-title')
    });
document.querySelector('.card-content').addEventListener('click', function() {
    console.log('card content');
});
document.querySelector('.card').addEventListener('click', function() {
    console.log('Card Itself');
});


/**
 * EVENT DELEGATION:
 *  --> The idea is that if we have a lot of elements handled in a similar way, then instead of assigning a handler to each of them – we put a single handler on their common ancestor.
 *  --> In the handler we get event.target, see where the event actually happened and handle it.
 *  --> Some Items are generated dynamically on the dom, if we need to target them and add event listeners, then we can use event delegation.
 */

 document.body.addEventListener('click', function(e) {
    if (e.target.parentElement.classList.contains('delete-item')) {
       e.target.parentElement.parentElement.remove();
    }
 });
```