Events:

1. Events represent important occurrences that have a time (the time when the event happened) and space (where the event happened)
2. Events are used to model stimulus that can make a state machine to change its state.
3. Events are some occurrences that can trigger the state transition of an object or a group of objects. Events have a location in time and space but do not have a time period associated with them.
4. The object that raises or triggers the event is the event sender, and the object that receives the event is the event receiver.
5. The event receiver does whatever it must do when that event occurs; however, there's no communication back to the object that triggered the event indicating that it was handled or the result if it was handled. Therefore, the event receiver is always a subroutine and not a function because the return value has no meaning.
6. Events are generally associated with some actions.
7. Examples of events are mouse click, key press, an interrupt, stack overflow, etc.
8. Events that trigger transitions are written alongside the arc of transition in state diagrams.
9. Events can be of two types -**synchronous or asynchronous**.
10. In a synchronous event the sender waits for an action from the receiver on the event. For example, pressing a button on the mouse, a phone call, etc.
11. There is no waiting in an asynchronous event. For example, sending an e-mail, letter, delivery of newspaper, etc.

**For Synchronous Events:**

* Sending or receiving is similar to call event
* The sender and the receiver are in a meeting (the sender sends the signal and wait for a response from the receiver) for the duration of the operation
* When an object calls an operation, the sender sends the operation and then waits for the receiver

**For Asynchronous Events:**

* Sending or receiving is similar to signal event
* The sender and receiver do not meet, i.e., the sender sends the signal but does not wait for a response from the receiver
* When an object sends a signal, the sender sends the signal and then continues along its flow of control, without waiting for any return from the receiver

1. Further Events can also be classified as:
2. **External events** are those events that pass from a user of the system to the objects within the system. For example, mouse click or key−press by the user are external events.
3. **Internal events** are those that pass from one object to another object within a system. For example, stack overflow, a divide error, etc.
4. **Deferred events** are those which are not immediately handled by the object in the current state but are lined up in a queue so that they can be handled by the object in some other state at a later time.

The kinds of events are

1. Signals: Signals refer to the objects that are dispatched asynchronously by one object and then received by another. Exceptions and e-mails are an example of internal signals. They may have instances, generalization relationships, attributes and operations. Attributes of a signal serve as its parameters. It may be sent as the action of a state transition in a state machine or the sending of a message in an interaction. Signals are modelled as stereotyped classes and the relationship between an operation and the events by using a dependency relationship, stereotyped as send
2. Calls: A call is a synchronous event representing the dispatch of an operation Example: Method calls in programming. It may trigger a state transition in a state machine or it may invoke a method on the target object
3. The passing of time: Time Event represents the passage of time. Modelled by using the keyword 'after' followed by some expression that evaluates to a period of time which can be simple or complex.
4. A change in state: Change event represents a change in state or the satisfaction of some condition. Modelled by using the keyword 'when' followed by some Boolean expression.

For Example:

The transition from waiting state to Riding state takes place when the person gets a taxi. Likewise, the final state is reached, when he reaches the destination. These two occurrences can be termed as events Get\_Taxi and Reach\_Destination.