

## State changes in objects

Just to give you an example, let us take water (H<sub>2</sub>O) as a type. In summer, mother nature heats up everything and the water in lakes, rivers, sea evaporates to become clouds in the sky. In winter, nature extracts heat from the water and it becomes ice to spread across fields, rivers, and mountains. Ice again becomes water and flows over mountains, plains to streams, and finally to sea, when mother nature completes the cycle by bringing the summer back.

What we have noticed from the above example?

We have definitely learned couple of things from it. A type, changes its state (in this case, the state is ice, water, or vapor). Something external that (in this case nature extracts or puts heat back to the water) acts to the object so that it moves from one state to another state.

Let us take the BankAccount class as an example. Assume that you have just opened a new bank account with \$100 as the opening deposit amount. We can call the state of your account is in create state. Once created an account might take some times before it can be operational, as the bank has to deliver an ATM or Credit card, and bunch of check to you before it becomes operational. Therefore, when creation state is over, you are ready to operate your account by depositing, withdrawing, check issuing against the account, called operational state.

Similarly, when you do not operate your account for a very long time and do not have enough balance in the account, bank might move the account to closed state and you'll not be able to operate your account. To operate the same account, you perhaps have to request the bank to re-open it to operational state. In terms of object definition, the state can be defined using just an integer flag to hold the three different values. Let us say 1 is for "open state"<sup>TM</sup>, 2 for "operational state"<sup>TM</sup> and 3 is to indicate "closed state"<sup>TM</sup>.