**constructor arguments**

The **constructor()** function executes when we create an object. This will allow us to create a box of our choice of coordinates.

In the **constructor()** function, we are going to pass the parameters here which will allow you to give custom parameters.

Here we are setting the parameters like **x**, **y**, **w**, **h,** and **vx,** which stands for x-position, y-position, width, height, and x speed of the box.

**constructor(x,y,w,h,vx)**

In the **setup()** function we create the object as :

**box1 = new Box(100,100,50,50,2)**.

In the **draw()** function, we call the class functions using objects as:

**box.show()**

**box.move()**

**Scope:**

Variables in the program have a scope. They live and die within the scope.

A variable declared inside a function or any block of code starting and ending with { } (for example - for loop) have scope defined only inside the function (or curly brackets).

It cannot be accessed by any other function or any other block of code directly. Such variables are said to have local scope.

Variables declared outside the functions at the top are said to have a global scope. This means you can access them anywhere in the code.

**mousePressedOver()** which detects a mouse pressed over any sprite.

It accepts the sprite as an argument and returns true if the mouse is pressed over the sprite and false otherwise.

In our game, our count value depends on the **frameCount**.

The frameCount goes on increasing even when the game is in END state.

Score resets and starts from 0 when the game is reset using **framerate**.

frameRate is the number of frames (images our game shows every second).

frameRate is nearly constant throughout the game and nearly equal to 60, depending on speed of CPU.or 30 in most cases.

We could write: score = score + Math.round(frameRate/60).

When we divide the frameRate with 60, we may get the result in the form of decimal numbers, but we want the score to be updated only with integers(1,2,3,4 etc).

In order to do that we can use the **Math.round()** function.

This function rounds off the value to the nearest integer.

For example if we have the number as 9.2, the round function will make it as 9 and give us an integer.

This will increase the count by 1 every frame.

When the game resets, the count becomes 0 and then the count starts from 0 again.