```
//Requisite header files for libraries used in code for communication and accelerometer
sensor in
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_LIS3DH.h>
#include <Adafruit_Sensor.h>

// Used for software SPI
#define LIS3DH_CLK 13
#define LIS3DH_MISO 12
#define LIS3DH_MOSI 11
// Used for hardware & software SPI
#define LIS3DH_CS 10

// software SPI
//Adafruit_LIS3DH lis = Adafruit_LIS3DH(LIS3DH_CS, LIS3DH_MOSI, LIS3DH_MISO,
LIS3DH_CLK);
// hardware SPI
//Adafruit_LIS3DH lis = Adafruit_LIS3DH(LIS3DH_CS);
// I2C
Adafruit_LIS3DH lis = Adafruit_LIS3DH();
 //Below are the column patterns for LEDs for each iteration for all characters involed in
displaying classes and dot pattern.
 int _[] = {0,0,0,0,0,0,0, 0,0,0,0,0,0,0, 0,0,0,0,0,0,0, 0,0,0,0,0,0,0, 0,0,0,0,0,0,0};
 int A[] = {0,0,1,1,1,1,1, 0,1,0,1,0,0,0, 1,0,0,1,0,0,0, 0,1,0,1,0,0,0, 0,0,1,1,1,1,1};
 int C[] = {0,1,1,1,1,1,0, 1,0,0,0,0,0,1, 1,0,0,0,0,0,1, 1,0,0,0,0,0,1, 1,0,0,0,0,0,1};
 int E[] = {1,1,1,1,1,1,1, 1,0,0,1,0,0,1, 1,0,0,1,0,0,1, 1,0,0,1,0,0,1, 1,0,0,1,0,0,1};
 int L[] = {1,1,1,1,1,1,1, 0,0,0,0,0,0,1, 0,0,0,0,0,0,1, 0,0,0,0,0,0,1, 0,0,0,0,0,0,1};
 int S[] = {0,1,1,0,0,0,1, 1,0,0,1,0,0,1, 1,0,0,1,0,0,1, 1,0,0,1,0,0,1, 1,0,0,0,1,1,0};
 int dot[]={1,1,0,0,0,0,0, 0,0,1,1,0,0,0, 0,0,0,0,1,1,0, 0,0,0,0,0,0,1, 0,0,0,0,0,0,0};

 int letterSpace;//variable for producing a space visual space between each adjacent
character
 int delayTime;//Key thing in POV display.This is the frequency for iteration shifting.
Calibrating this time will improve POV display.Its in milliseconds(ms).

 //This is setup function in which all initial parameters are set and pin modes are
selected(output or input)\
 //The setup function executes only once when Arduino resets.
 void setup()
 {
         Serial.begin(9600);
         while (!Serial) delay(10);     // Initial pause for 10ms until serial module initializes.

         Serial.println("LIS3DH test!");
```

```
        if (! lis.begin(0x18)) {   // communication address for the sensor.(Its identity)
          Serial.println("Couldnt start");//If sensor doesnt respond then this is printed on
serial monitor
          while (1) yield();
        }
        Serial.println("LIS3DH found!");//This is printed if sensor responds properly.

        lis.setRange(LIS3DH_RANGE_4_G);   // 2, 4, 8 or 16 G!

        Serial.print("Range = "); Serial.print(2 << lis.getRange());
        Serial.println("G");
       for( int i = 2; i<9 ;i++ ) // setting the ports of the leds to OUTPUT
         {
          pinMode(i, OUTPUT);
         }
            //Initializing these two variables.
       letterSpace =4;// defining the space between the letters (ms)
       delayTime =1;// defining the time dots appear (ms)
}

//This is the subroutine/function/method created which takes an integer array as input
and moves through its index one by one thus lighting up leds.
 void printLetter(int letter[])
 {
         int y;//Local variable used for looping.
         //Displaying first column of character matrix.
         for (y=0; y<7; y++)
         {
         digitalWrite(y+2, letter[y]);
         }
         delay(delayTime);//Rate how fast the led's change their state to next column.
         //2nd column
         for (y=0; y<7; y++)
         {
         digitalWrite(y+2, letter[y+7]);
         }
         delay(delayTime);
         //3rd column
         for (y=0; y<8; y++)
         {
         digitalWrite(y+2, letter[y+14]);
         }
         delay(delayTime);
         //4rth column
         for(y = 0; y<7; y++) {
         digitalWrite(y+2, letter[y+21]);
```

```
           }
           delay(delayTime);
           //5th column
          for(y = 0; y<7; y++) {
           digitalWrite(y+2, letter[y+28]);
           }
           delay(delayTime);
           //displaying letter space turning all leds off.
           for (y=0; y<7; y++)
           {
           digitalWrite(y+2, 0);
           }
           delay(letterSpace);
 }
 void loop()
 {
    lis.read();      // get X Y and Z data at once
  // Then print out the raw data
  Serial.print("X:  "); Serial.print(lis.x);
  Serial.print("  \tY:  "); Serial.print(lis.y);
  Serial.print("  \tZ:  "); Serial.print(lis.z);

  /* Or....get a new sensor event, normalized */
  sensors_event_t event;
  lis.getEvent(&event);

  /* Display the results (acceleration is measured in m/s^2) */
  Serial.print("\t\tX: "); Serial.print(event.acceleration.x);
  Serial.print(" \tY: "); Serial.print(event.acceleration.y);
  Serial.print(" \tZ: "); Serial.print(event.acceleration.z);
  Serial.println(" m/s^2 ");

  Serial.println();
  if(event.acceleration.y>=3)//if there is 3m/s^2 acceleration in positive y axis of sensor,
then display "classes" character by character.
  {
     printLetter (C);
     printLetter (L);
     printLetter (A);
     printLetter (S);
     printLetter (S);
     printLetter (E);
     printLetter (S);

  }
  else{//otherwise keep printing dots in a swinging pattern.
  printLetter (dot);
  }
}
```