```
#include <MIDI.h>
#include <Servo.h>

MIDI CREATE DEFAULT INSTANCE();

Servo Servo1;  // First rotational servo  Notes G5-79, A5-81, B5-83, C6-84, D6-86, E6-88, F6-89, G6-91  (-xx = midi note number)
Servo Servo2;  // First hammer servo
Servo Servo3;  // Second rotational servo  Notes G#5-80, Bb5-82, C#6-85, D#6-87, F#6-90, G#6-92
Servo Servo4;  // Second hammer servo
Servo Servo5;  // Third rotational servo  Notes A6-93, B6-95, C7-96, D7-98, E7-100, F7-101, G7-103
Servo Servo6;  // Third hammer servo
Servo Servo7;  // Fourth rotational servo  Notes Bb6-94, C#7-97, D#7-99, F#7-102
Servo Servo8;  // Fourth hammer servo

//Note array {rotation angle, rotational servo number, repeated for every note in chromatic order starting on midi note 79 thru 103
byte angles[50]={44,1,128,3,52,1,120,3,60,1,68,1,108,3,76,1,101,3,85,1,91,1,87,3,100,1,81,3,65,5,119,7,72,5,78,5,106,7,84,5,99,7,89,5,96,5,87,7,101,5};
byte note;

// --------------------------------------------------------------------------

// This function will be automatically called when a NoteOn is received.
// It must be a void-returning function with the correct parameters,

void handleNoteOn(byte channel, byte pitch, byte velocity)
{
  note = (pitch-79)*2;            //calculates the array pointer to the angle for the given midi pitch value
  if(angles[note+1]==1){          //checks the next array value to see whether the pitch is for first rotational servo number 1
    Servo1.write(angles[note]);   //if yes rotate to the correct angle
    delay(250);                   //wait 250ms for the servo to travel into position
    Servo2.write(91);             //lower first hammer servo 2
    delay(50);                    //delay 50ms
    Servo2.write(85);             //raise first hammer servo 2
  }
  if(angles[note+1]==3){          //checks the next array value to see whether the pitch is for second rotational servo number 3
    Servo3.write(angles[note]);   //if yes rotate to the correct angle
    delay(250);                   //wait 250ms for the servo to travel into position
    Servo4.write(92);             //lower second hammer servo 4
    delay(50);                    //delay 50ms
    Servo4.write(99);             //raise second hammer servo 4
  }
  if(angles[note+1]==5){          //checks the next array value to see whether the pitch is for third rotational servo number 5
    Servo5.write(angles[note]);   //if yes rotate to the correct angle
    delay(250);                   //wait 250ms for the servo to travel into position
    Servo6.write(93);             //lower third hammer servo 6
    delay(50);                    //delay 50ms
    Servo6.write(88);             //raise third hammer servo 6
  }
  if(angles[note+1]==7){          //checks the next array value to see whether the pitch is for fourth rotational servo number 7
    Servo7.write(angles[note]);   //if yes rotate to the correct angle
    delay(250);                   //wait 250ms for the servo to travel into position
    Servo8.write(94);             //lower fourth hammer servo 8
    delay(50);                    //delay 50ms
    Servo8.write(102);             //raise fourth hammer servo 8
  }

}

//Note off commands are ignored as a glockenspiel does not need it
void handleNoteOff(byte channel, byte pitch, byte velocity)
{
    // Do something when the note is released.
    // Note that NoteOn messages with 0 velocity are interpreted as NoteOffs.
```

```
}

// ------------------------------------------------------------------------------


void setup()
{                   //pin numbers used for the servos
  Servo1.attach(2); // attaches the servo on pin 2 to the servo object
  Servo2.attach(3); // attaches the servo on pin 3 to the servo object
  Servo3.attach(4); // attaches the servo on pin 4 to the servo object
  Servo4.attach(5); // attaches the servo on pin 5 to the servo object
  Servo5.attach(6); // attaches the servo on pin 6 to the servo object
  Servo6.attach(7); // attaches the servo on pin 7 to the servo object
  Servo7.attach(8); // attaches the servo on pin 8 to the servo object
  Servo8.attach(9); // attaches the servo on pin 9 to the servo object
  MIDI.setHandleNoteOn(handleNoteOn);     //declare note on handler
  MIDI.setHandleNoteOff(handleNoteOff);   //declare note off handler
  MIDI.begin(MIDI CHANNEL OMNI);          //begin midi object looking at all midi channels
  Serial.begin(115200);                   //use computer baud rate not the true midi baud rate of 31250
}

void loop()
{                         //initial servo angles
  Servo1.write(68);       //first rotational servo angle C6
  Servo2.write(85);       //first hammer up
  Servo3.write(108);      //second rotational servo angle C#6
  Servo4.write(99);       //second hammer up
  Servo5.write(84);       //third rotational servo angle D7
  Servo6.write(88);       //third hammer up
  Servo7.write(106);       //fouth rotational servo angle C#7
  Servo8.write(102);       //fourth hammer up
  delay(1000);            //wait 1s before starting

  while(true){            //continuous loop read the serial port for midi commands
    MIDI.read();
  }

}
```