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

#define SERVO_PIN 3
#define ULTRASONIC_SENSOR_TRIG 11
#define ULTRASONIC_SENSOR_ECHO 12
#define MAX_REGULAR_MOTOR_SPEED 200
#define MAX_MOTOR_ADJUST_SPEED 200
#define DISTANCE_TO_CHECK 45

//Right motor
int enableRightMotor=5;
int rightMotorPin1=7;
int rightMotorPin2=8;

//Left motor
int enableLeftMotor=6;
int leftMotorPin1=9;
int leftMotorPin2=10;

NewPing mySensor(ULTRASONIC_SENSOR_TRIG, ULTRASONIC_SENSOR_ECHO, 400);
Servo myServo;
void setup()
{
  // put your setup code here, to run once:
  pinMode(enableRightMotor,OUTPUT);
  pinMode(rightMotorPin1,OUTPUT);
  pinMode(rightMotorPin2,OUTPUT);

  pinMode(enableLeftMotor,OUTPUT);
  pinMode(leftMotorPin1,OUTPUT);
  pinMode(leftMotorPin2,OUTPUT);

  myServo.attach(SERVO_PIN);
  myServo.write(90);
  rotateMotor(0,0);
}

void loop()
{

  int distance = mySensor.ping_cm();

  //If distance is within 30 cm then adjust motor direction as below
  if (distance > 0 && distance < DISTANCE_TO_CHECK)
  {
    //Stop motors
    rotateMotor(0, 0);
    delay(500);

    //Reverse motors
    rotateMotor(-MAX_MOTOR_ADJUST_SPEED, -MAX_MOTOR_ADJUST_SPEED);
    delay(200);

    //Stop motors
    rotateMotor(0, 0);
    delay(500);

    //Rotate servo to left
    myServo.write(180);
    delay(500);

    //Read left side distance using ultrasonic sensor
    int distanceLeft = mySensor.ping_cm();

    //Rotate servo to right
    myServo.write(0);
    delay(500);

    //Read right side distance using ultrasonic sensor
    int distanceRight = mySensor.ping_cm();

    //Bring servo to center
    myServo.write(90);
    delay(500);

    if (distanceLeft == 0 )
    {
      rotateMotor(MAX_MOTOR_ADJUST_SPEED, -MAX_MOTOR_ADJUST_SPEED);
      delay(200);
    }
    else if (distanceRight == 0 )
    {
      rotateMotor(-MAX_MOTOR_ADJUST_SPEED, MAX_MOTOR_ADJUST_SPEED);
      delay(200);
    }
    else if (distanceLeft >= distanceRight)
    {
      rotateMotor(MAX_MOTOR_ADJUST_SPEED, -MAX_MOTOR_ADJUST_SPEED);
      delay(200);
    }
    else
    {
      rotateMotor(-MAX_MOTOR_ADJUST_SPEED, MAX_MOTOR_ADJUST_SPEED);
      delay(200);
    }
    rotateMotor(0, 0);
    delay(200);
  }
  else
  {
    rotateMotor(MAX_REGULAR_MOTOR_SPEED, MAX_REGULAR_MOTOR_SPEED);
  }

}

void rotateMotor(int rightMotorSpeed, int leftMotorSpeed)
{
  if (rightMotorSpeed < 0)
  {
    digitalWrite(rightMotorPin1,LOW);
    digitalWrite(rightMotorPin2,HIGH);
  }
  else if (rightMotorSpeed >= 0)
  {
    digitalWrite(rightMotorPin1,HIGH);
    digitalWrite(rightMotorPin2,LOW);
  }

  if (leftMotorSpeed < 0)
  {
    digitalWrite(leftMotorPin1,LOW);
    digitalWrite(leftMotorPin2,HIGH);
  }
  else if (leftMotorSpeed >= 0)
  {
    digitalWrite(leftMotorPin1,HIGH);
    digitalWrite(leftMotorPin2,LOW);
  }

  analogWrite(enableRightMotor, abs(rightMotorSpeed));
  analogWrite(enableLeftMotor, abs(leftMotorSpeed));
}
```