```
/*
 * RBE 1001 Sensors homework
 *
 * Program: Control a servo using a potentiometer. As the pot is moved from stop-to-stop the
 *  servo should be moved through its full range of motion.
 *
 * The following code is based on the "Knob" program provided as part of the
 * ESP32 Arduino examples.
 *
 * The pot input (wiper) is connected to pin 34; the servo is connected to digital I/O pin 17.  *
 *
Modified by: Craig B. Putnam, 4/4/16 and Brad Miller 9/22/17, and Nick Bertozzi, 11/11/19
 */

#include <ESP32Servo.h>

Servo myservo;                      // create servo object to control a servo

const int servoPin = 17;   // specify the I/O pin for the servo
const int potPin = 34;     // specify the I/O pin for the pot

void setup() {
  myservo.attach(servoPin);          // attach the servo on the specified pin
}

// read the analog pin which is a value from 0-4095 representing values from 0-3.3V
// and scale it to a value from 0-180. The assignment specifically said to not use
// the arduino map() function, which is normally very convenient for these types of
problems. // If it were allowed, it would look like this: map(val, 0, 4095, 0, 180) rather
// than doing the math shown below.
// In the expression: val * 180 / 4095 it is important to do the multiply before
// dividing by 4095 because these are integer values and doing the division first (even
// though it's more natural) would not work because it would make a number that is
// a fraction from 0-1 that would get truncated to 0. This is because the values are
// integers and not floating point numbers.
void loop() {
   long val = analogRead(potPin);      // reads the value of the potentiometer (value between 0
and 4095)
                           // scale it to use it with the servo (value between 0 and 180)
   myservo.write(val * 180/4095);    // sets the servo position according to the scaled value
   delay(15);                         // waits for the servo to get there
}

// to flip the direction of the servo, subtract the resultant 0-180 value from 180
//      180 - val * 180 / 4095
// or if you could use the map function then it would look like this:
//      map(val, 0, 4095, 180, 0);
// which would change 0 to 180 and 180 to 0.
```