**Arduino Programming Assignment (20 Questions)**

Ideal for: Students, hobbyists, or job candidates applying for embedded or IoT-related roles.

**Section 1: Basics (Q1–Q5)**

1. **Blink an LED**  
   Write a program to blink the built-in LED (pin 13) every 1 second.
2. **Digital Read**  
   Read a button input connected to pin 2 and turn on the LED if the button is pressed.
3. **PWM Control**  
   Use analogWrite() to dim an LED on pin 9 in a loop from 0 to 255 and back.
4. **Debounce a Button**  
   Implement a simple debounce algorithm for a button that toggles an LED.
5. **State Toggle**  
   Every time the button is pressed, toggle the LED state (on/off) using only one button.

**Section 2: Sensors & Inputs (Q6–Q10)**

1. **Analog Read**  
   Read an analog value from a potentiometer and display it on the Serial Monitor.
2. **Temperature Sensor**  
   Connect an LM35 sensor and print the temperature in Celsius to the Serial Monitor.
3. **Light Sensor Trigger**  
   Turn on an LED when the LDR (photoresistor) value drops below a threshold.
4. **Joystick Control**  
   Read X and Y axis values of a joystick and print direction (LEFT/RIGHT/UP/DOWN) to Serial Monitor.
5. **Distance Measurement**  
   Interface an HC-SR04 ultrasonic sensor and print the distance in cm.

**Section 3: Outputs & Actuators (Q11–Q14)**

1. **Servo Motor Sweep**  
   Use the Servo library to sweep a servo from 0 to 180 degrees and back.
2. **Buzzer Alarm**  
   Play a warning tone when a button is pressed using a piezo buzzer.
3. **Traffic Light Simulation**  
   Create a traffic light sequence using three LEDs (red, yellow, green).
4. **Seven Segment Display**  
   Display numbers 0–9 in a loop on a 7-segment display using digital pins.

**Section 4: Advanced Concepts (Q15–Q17)**

1. **Interrupts**  
   Use an external interrupt (pin 2 or 3) to increment a counter each time a button is pressed.
2. **EEPROM Read/Write**  
   Save a value to EEPROM on button press and retrieve it on reset.
3. **Timer with millis()**  
   Blink an LED without using delay(), using millis() instead for timing.

**Section 5: Communication & Integration (Q18–Q20)**

1. **Serial Communication**  
   Receive a number from Serial Monitor and blink the LED that many times.
2. **I2C Communication**  
   Connect and display temperature data on an I2C-based OLED display.
3. **Bluetooth Control**  
   Control an LED from a mobile app via HC-05 Bluetooth module commands (e.g., "ON", "OFF").

**Instructions for Candidate**

* Use the Online Arduino IDE or PlatformIO to write,compile and upload your code.

<http://arduinodev.com/software/builder/>

<https://onecompiler.com/cpp/3ygvuaj5c>

Simulations :

<https://wokwi.com/arduino>

* Include comments in your code to explain logic.
* Submit .ino files (or .zip project folder) along with wiring diagrams (optional).
* Focus on code **readability**, **reusability**, and **hardware understanding**.