ARDUINO BEGINNER COURSE

Week 1: Introduction to Arduino

Lesson 1: Overview of Arduino and Setting Up

- Understanding the components of the Arduino board.
- Installing and setting up the Arduino IDE.
- Introduction to basic coding structure (setup() and loop() functions).

Project: Blink the Built-in LED

- Blink the built-in LED on the Arduino board using the digitalWrite() function to get familiar with the board.

Lesson 2: Understanding the Breadboard and Simple Circuits

- How a breadboard works.
- Wiring LEDs and resistors.
- Push button introduction.
- Writing simple code to control circuits.

Project: Control an LED with a Button

- Make an LED light up when pressing a button using the digitalRead().

Week 2: Basic Coding for Arduino

Lesson 3: Variables and Functions in Arduino

- Understanding variables and data types in Arduino.
- Writing and using functions in the setup() and loop() sections.
- Basic logic with if-else statements.

Project: LED Pattern with Variables

- Create a blinking pattern using multiple LEDs, controlled by variables and functions.

Lesson 4: Loops and Control Structures

- Introduction to control structures (for, while loops).
- How to use loops to repeat tasks.
- Implementing if-else conditions for decision-making.

Project: Counting Button Presses

- Connecting a push button and an LED then use the loops to count the button presses.

Week 3: Serial Communication

Lesson 5: Introduction to Serial Communication

- How to use the Serial Monitor for debugging.
- Sending data from Arduino to a computer.
- Using the Serial.print() function to display sensor data.

Project: Displaying some texton the Serial Monitor

- Use the Serial Monitor to display some text, variables or LED status.

Lesson 6: Sending Commands to Arduino via Serial

- Sending commands from the Serial Monitor to control Arduino components.
- Reading and interpreting Serial input data.
- Controlling LEDs or other components based on Serial input.

Project: Serial-Controlled LED

- Control an LED on/off state by typing commands into the Serial Monitor.

Week 4: Using LEDs and RGB LEDs

Lesson 7: Controlling LEDs with Arduino

- Understanding digital output pins.
- Controlling multiple LEDs using the Arduino.
- Exploring basic delay() functions for timing.

Project: LED Traffic Light Simulation

- Create a simple traffic light system using loops and control structures for multiple LEDs.

Lesson 8: RGB LEDs

- How RGB LEDs work.

- Using analogWrite() to control the brightness of LEDs.
- Combining different colors with RGB LEDs.

Project: Color Fader

- Create a simple RGB color fader using analogWrite() to mix different colors over time.

Week 5: Introduction to Sensors

Lesson 9: Overview of Sensors

- What are sensors?
- How they work with Arduino.
- Introduction to different types of sensors (temperature, distance, light).

LM35 Temperature Sensor Introduction

- Understanding how temperature sensors work.
- Connecting the LM35 sensor to Arduino.
- Reading analog values and converting them to temperature.

Project: Temperature Display

- Measure the room temperature and display it using the Serial Monitor.

Lesson 10: Temperature Sensor with RGB LED

- How temperature sensors work.
- Using the LM35 temperature sensor with Arduino.
- Integrating the RGB LED for visual temperature indication.

Project: RGB Temperature Indicator

- Use an LM35 temperature sensor to change the color of an RGB LED based on temperature readings (e.g., red for hot, green for normal, blue for cold).

Week 6: Servo Motors and Motion Control

Lesson 11: Basics of Servo Motors

- Understanding how servo motors work.

- Controlling servo position with Arduino.
- Writing code to set servo angles using Servo library.

Project: Servo Sweeper

- Make the servo motor sweep back and forth between two angles.

Lesson 12: Advanced Servo Control

- Using potentiometers to control servo motors.
- Reading analog input and mapping values to control motion.

Project: Servo Position Control with Potentiometer

- Use a potentiometer to control the position of a servo motor in real-time.

Week 7: Buzzer and Sound Applications

Lesson 13: Introduction to Buzzers and Sound

- Learn how to use a buzzer to create sound based on sensor input.
- Students understand how to control sound output with Arduino.

Project: Program the Arduino to generate a tone using a buzzer when a button is pressed.

Lesson 14: Creating a Simple Alarm System

- Build a simple alarm system using an ultrasonic sensor and buzzer.
- Students build a practical project combining sensors and buzzers.

Project: Create a alarm program that triggers the buzzer when an object is detected within a certain distance.

Week 8: Ultrasonic Sensors

Lesson 15: Ultrasonic Sensors for Distance Measurement

- Introduction to the ultrasonic sensor.
- Reading distance data from the sensor.
- Calculating and converting raw data into meaningful distances.

Project: Distance-Based Light Control

- Use an ultrasonic sensor to turn an LED on/off depending on the proximity of an object.

Lesson 16: Advanced Distance-Based Projects

- Using distance data to control other components.
- Implementing conditions and actions based on distance thresholds.

Project: Parking Assistance System

- Create a parking assistant project using the ultrasonic sensor to measure the distance and provide a warning when the car is too close to an obstacle (using LEDs or buzzers).

Week 9: Components Intergrations and Personal Projects Planning

Lesson 17: Integrating Multiple Components

- How to integrate LEDs, sensors, and buzzers into one project.
- Exploring multi-component system interactions.

Project: LED Alarm System

- Combine sensors, buzzers, and LEDs into a functional alarm system.

Lesson 18: Project Brainstorming and Documentation

- Group brainstorming for personal projects.
- Identifying components and objectives for each project.
- Introduction to creating a project plan.

Activity: Students plan their personal projects and create a basic project documentation, detailing objectives, components, and expected outcomes.

Week 10: Debugging and Refining Projects

Lesson 19: Debugging and Troubleshooting

- Common errors in components, programming and how to fix them.
- Troubleshooting wiring and sensor issues.

Project: Debugging Challenge

- A pre-built project with intentional errors for students to debug and fix.

Lesson 20: Quiz and Q&A Session

- A quiz to review and assess knowledge from the course.
- Q&A session for students to ask questions about programming, components or circuits.

Week 11: Personal Project Development

Lesson 21: Personal Project Development - Part 1

- Hands-on project development.
- Guidance on circuit design, code structure, and troubleshooting.

Lesson 22: Personal Project Development - Part 2

- Continued work on personal projects.
- Teachers provide support and feedback on progress.

Week 12: Final Project Presentations

Lesson 23: Project Presentations - Part 1

- Half the class presents their projects.
- Demonstrations of functionality and explanations of the process.

Lesson 24: Project Presentations - Part 2

- Remaining students present their projects.
- Continued demonstrations and discussions.

Week 13: Reflection and Wrap-Up

Lesson 25: Feedback and Reflection

- Reflection on challenges, successes, and lessons learned.
- Group discussion on what worked and areas for improvement.

Lesson 26: Certification and Course Wrap-Up

- Celebrate achievements and issue course certificates.
- Encourage students to continue exploring Arduino and other technologies.