MOONPRENEUR
Future-proofing the Next Generation

| Tier-3 Robotics | |
|---|---|
| **Session No** | **Course Contents** |
| **Session 1** | **Introduction to Python**<br>**1. Python Introduction –**<br>Why Python is used in robotics, AI, and automation.<br>**2. Python vs. C++ –**<br>Simple syntax, no data type declaration, uses indentation.<br>**3. Real-Life Uses –**<br>Python powers apps like Google, Tesla, Netflix.<br>**4. Installation & Setup –**<br>How to install Python and write the first"Hello, World!" code. |
| **Session 2** | **Code That Listens Inputs, Variables & Datatypes**<br>**1. User Input & Variables –**<br>Learning how to take input using input() and store it in variables.<br>**2. Data Types –**<br>Understanding basic Python data types like int, float, str, bool, list, and dict.<br>**3. ID Card Generator Project –**<br>Creating a student ID card program using variables and input.<br>**4. Math Bot Program –**<br>Building a Python program to perform addition, subtraction, and multiplication.<br>**5. Difference from Arduino –**<br>Comparing Python's flexibility with<br>Arduino C++ regarding variables and inputs. |
| **Session 3** | **Calculator Using Python**<br>**1. Python Calculator Project –**<br>Building a calculator that performs addition, subtraction, multiplication, and division using if-elif-else conditions.<br>**2. Error Handling (Division by Zero) –** Learning how to handle division by zero using nested if statements.<br>**3. Loops with while True –**<br>Making the calculator run continuously until the user types "exit".<br>**4. Indentation Importance –**<br>Understanding Python's indentation rules compared to Arduino C++ syntax. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

MOONPRENEUR
Future-proofing the Next Generation

| | |
|---|---|
| **Session 4** | **MicroPython Adventures with Pi Pico**<br>**1. Introduction to Raspberry Pi Pico & Wokwi Simulator–**<br>Differences between Arduino Uno and Pi Pico in terms of microcontroller, memory, GPIO pins, and programming language.<br>**2. On-board LED Blinking Using MicroPython–**<br>Writing code to blink Pi Pico's onboard LED using Pin(25, Pin.OUT) with while True loop.<br>**Understanding MicroPython Syntax vs. Arduino C++–**<br>Comparison of pin setup, LED control, delay, and loop structure between MicroPython and Arduino C++.<br>**4. Controlling an External LED with Pi Pico–**<br>Replacing onboard LED with an external LED connected to GPIO pin 0, using the same blinking logic. |
| **Session 5** | **Lighting Logic**<br>**1. Forward LED Chasing Pattern Using For Loop–** Students learned how to use a for loop with range(1, 6) to turn on LEDs from GP1 to GP5 in sequence using MicroPython.<br>**2. Understanding range() Function and For Loop Syntax–**<br>Explained how start, stop, and step values work in Python's range() and compared it with C++ loop structure.<br>**3. Reverse LED Chasing Pattern Concept–**<br>Introduced using range(4, 1, -1) for reverse movement of LEDs, making lights chase back from GP4 to GP2.<br>**4. For Loop Comparison: Python vs Arduino C++ –**<br>Discussed syntax differences like indentation in Python vs. curly braces in C++, automatic vs. manual loop variable declarations.<br>**5. Loop-Based LED Control with Sleep Delays–**<br>Reinforced the use of sleep () function for delay timing to control how long each LED stays on and off in the chasing effect. |
| **Session 6** | **Exploring Pi Pico with LED Bar Graph**<br>**1. Understanding LED Bar Graph and Single LED Blinking –**<br>Students learned what an LED bar graph is and implemented a basic blinking program using GP10 with a 0.5-second delay to understand GPIO control in MicroPython. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

| | |
|---|---|
| | **2. Simulating Fuel Filling Using Growing LED Bar –** Explored how to light up LEDs one by one from GP10 to GP1 using a loop and range(10, 0, -1) to simulate a tank filling effect, introducing visual progress through LED sequencing.<br>**3. GPIO and ADC Comparison: Raspberry Pi Pico vs Arduino Uno –** Compared hardware specs like GPIO pins, analog inputs, clock speed, RAM, flash memory, and ADC resolution (12-bit vs 10-bit) to understand hardware capabilities.<br>**4. Controlling LED Bar with Button Input –** Implemented logic where each press of a button turns on the next LED in the bar from GP10 to GP1, using a counter variable and dynamic pin assignment. |
| **Session 7** | **Potentiometer Palette: Mix Your MoodLight**<br>**1. RGB Mood Light Using Potentiometer and Button –** Students created a color mixer using an RGB LED controlled via PWM signals, with brightness for each color (Red, Green, Blue) set using a potentiometer and confirmed using a push button.<br>**2. Using ADC to Read Potentiometer Values –** Explored how the Pi Pico's ADC reads analog values (0–65535) and how those values are scaled to 0–255 for RGB control, demonstrating analog-to-digital conversion in real time.<br>**3. PWM Frequency and Duty Cycle Control –** Understood how PWM (1000 Hz) is used to set LED brightness by converting RGB values back to a 16-bit range (0–65535), affecting LED brightness via duty_u16().<br>**4. Interactive Color Setting Logic –** Implemented a loop that lets users adjust Red, Green, and Blue intensities one by one, confirm each via a button press, and then view the mixed color result on the RGB LED |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

| | |
|---|---|
| **Session 8** | **The Dodger(LCD game)**<br>**1. Custom Character Display on LCD –**<br>Students learned to interface a 16x2 LCD with Raspberry Pi Pico and display custom characters like a walking man and an obstacle using a custom lcd.py module.<br>**2. Building a Dodger Game with Obstacle Movement –**<br>Developed a simple game where a player character stays in a fixed column while obstacles randomly move from right to left across two rows of the LCD.<br>**3. Player Control Using Push Button –**<br>Implemented button-based control to toggle the player's row (top or bottom) to dodge incoming obstacles, introducing the concept of toggling state with input.<br>**4. Collision Detection and Scoring Mechanism –**<br>Added logic to detect collisions between the player and obstacle, reset the game upon collision, and introduced a scoring system that increments each time the obstacle safely passes the player. |
| **Session 9** | **Time and Tones with MicroPython**<br>**1. Controlling a Buzzer with PWM –**<br>Students learned to use Pulse Width Modulation (PWM) on GPIO 13 to turn a buzzer ON and OFF, adjusting frequency and duty cycle to generate sound.<br>**2. Understanding Duty Cycle and Frequency –**<br>Explained how a 50% duty cycle (32768) creates medium-volume sound, and how changing the frequency (in Hz) changes the pitch of the buzzer tone.<br>**3. Using ticks_ms() to Track Time –**<br>Introduced the ticks_ms() function from the time module to measure elapsed time in milliseconds, enabling timed control of sound output.<br>**4. Creating a Timed Melody –**<br>Implemented a program that plays four musical notes (C4, E4, G4, A4) at specific time intervals (0–8 seconds), then automatically stops the buzzer after the melody ends. |
| **Session 10** | **Show & Tell:**<br>Students showcase their projects, demonstrating creativity and learning. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

MOONPRENEUR
Future-proofing the Next Generation

| | |
|---|---|
| **Session 11** | **Kit Unboxing & Verification**<br>Unboxing the kit, verifying components, and understanding their functions. |
| **Session 12** | **The Motor Mechanics: Understanding DC motors**<br>**L**earn about motors, assembly of motors to the robot, and interfacing it to make the robot provide motion |
| **Session 13** | **Basics of Robotic Navigation**<br>Understanding robotic navigation, movement challenges, and turn types. |
| **Session 14** | **Assembly of I2C LCD:**<br>Assembling and interfacing an I2C LCD to display sensor data. |
| **Session 15** | **Manual Controlled Robot:**<br>Controlling the robot using serial inputs and manual keyboard commands. |
| **Session 16** | **Limit Switch:**<br>Using limit switches to detect collisions and stop the robot automatically. |
| **Session 17** | **Limit Switch - II:**<br>Enhancing collision detection with limit switches for speed variation. |
| **Session 18** | **Introduction to IR Sensor:**<br>Introduction to IR sensors and their role in obstacle detection. |
| **Session 19** | **Inventory Tracker using IR:**<br>Using IR sensors to build an inventory tracker that counts objects. |
| **Session 20** | **Show & Tell Mock:**<br>Students prepare and present their show n tell project as a final mock and make the final changes. |
| **Session 21** | **Show & Tell:**<br>Students showcase their projects, demonstrating creativity and learning. |
| **Session 22** | **Interfacing Servo Motor:**<br>Interfacing and controlling a servo motor for directional movement. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

MOONPRENEUR
Future-proofing the Next Generation

| | |
|---|---|
| **Session 23** | **Interfacing Ultrasonic Sensor:** Assembling an ultrasonic sensor and using it for precise distance measurement. |
| **Session 24** | **Autonomous Obstacle Avoiding Robot:** Building an autonomous obstacle-avoiding robot using ultrasonic sensors. |
| **Session 25** | **Path Finder** Creating a pathfinder robot that scans and navigates around obstacles. |
| **Session 26** | **Functions that talk back** Exploring functions with return types and default parameters in Arduino. |
| **Session 27** | **One Pin, Two Directions: The Magic of Inverter Mode** Introducing inverter mode to simplify motor direction control with one pin. |
| **Session 28** | **Custom Library** Learning how to create and use custom Arduino libraries for modular code. |
| **Session 29** | **RoboCop** Building a RoboCop surveillance robot that scans and detects intruders. |
| **Session 30** | **Show & Tell Mock** Students prepare and present their show n tell project as a final mock and make the final changes. |
| **Session 31** | **Show & Tell** Students showcase their projects, demonstrating creativity and learning. |
| **Session 32** | **Object Following Robot** Developing an object-following robot that tracks movement dynamically. |
| **Session 33** | **Introduction to Accelerometer** Introduction to accelerometers and their real-world robotic applications. |
| **Session 34** | **Accelerometer - II** Using accelerometers to detect sudden hits and surface inclinations. |
| **Session 35** | **Introduction of IR Array** Understanding IR array sensors and their role in robotic applications. |
| **Session 36** | **Line follower Robot Part 1** Building a basic line follower robot using two IR sensors. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.

MOONPRENEUR
Future-proofing the Next Generation

| | |
|---|---|
| **Session 37** | **Line follower Robot Part 2** Improving the line follower robot with additional sensors and functions. |
| **Session 38** | **Line follower Robot Part 3** Advanced line-following techniques for more accurate path tracking. |
| **Session 39** | **Show & Tell Mock** Students prepare and present their show n tell project as a final mock and make the final changes. |
| **Session 40** | **Show & Tell** Students showcase their projects, demonstrating creativity and learning. |
| **Session 41** | **Design Thinking** Applying design thinking to solve real-world robotic challenges. |
| **Session 42** | **Introduction** Introduction to environmental sensing and real-world applications. |
| **Session 43** | **Measuring the Temperature and Humidity** Measuring temperature and humidity using digital sensors. |
| **Session 44** | **Air Quality Monitoring** Monitoring air quality using gas sensors to detect pollutants. |
| **Session 45** | **Introduction to PPM** Understanding PPM (Parts Per Million) and its role in air quality measurement. |
| **Session 46** | **Wireless Control** Implementing wireless control for real-time robotic movement. |
| **Session 47** | **Wireless Robot** Building a fully functional wireless-controlled robot.. |
| **Session 48** | **Menu - Selecting the Mode** Creating a menu system to select different robot operating modes. |
| **Session 49** | **Data Collection** Collecting and analyzing data from robotic sensors for decision-making. |
| **Session 50** | **Menu Optimization** Optimizing the menu system for better user experience and functionality. |
| **Session 51** | **Wirelessly Switching Modes** Switching between different robot modes wirelessly with ease. |
| **Session 52** | **Final Show n Tell** Final project showcase where students display their complete robotics journey. |

**Disclaimer**

** 1. The curriculum is subjected to changes and upgradation as per the feedback received from our customers.
2. Project and activities will depend on the type of course and tier.
3. Moonpreneur reserves all the rights to update its curriculum at any given point in time.