Note: The duration of each lesson can be adjusted based on the available time and the pace of the students.

Note: The code examples provided are simplified and may need to be adapted based on the specific hardware and libraries used in the Donkey Car project.

Week 1: Introduction to Donkey Car and AI

- Introduction (5 minutes)
  - Welcome students and provide an overview of the lesson plan.
  - Explain the importance of AI in autonomous vehicles and its applications.

- What is Donkey Car? (10 minutes)
  - Explain the concept of Donkey Car as an open-source DIY self-driving platform.
  - Discuss the various components of a Donkey Car, such as the chassis, motor, camera, and microprocessor.

- Introduction to AI (15 minutes)
  - Define AI and its applications in various fields.
  - Discuss the role of AI in autonomous vehicles and its importance in Donkey Car.

- Activity: AI Examples (20 minutes)
  - Provide examples of AI applications in everyday life, such as voice assistants, recommendation systems, and image recognition.
  - Discuss how AI can be applied to Donkey Car to enable autonomous driving.

- Recap and Homework (10 minutes)
  - Summarize the key points covered in the lesson.
- Assign homework: Research and find one real-world example of AI in autonomous vehicles.

Week 1: Introduction to Donkey Car and AI

Python Code Example:
```python
# Example Python code for controlling Donkey Car

# Import necessary libraries
import time

# Define a function to drive the car forward
def drive_forward():
    # Code to control the motors and drive the car forward
    print("Driving forward...")

# Call the function to drive the car forward
drive_forward()
```

MicroPython Code Example:
```python
# Example MicroPython code for controlling Donkey Car

# Import necessary libraries
import time

# Define a function to drive the car forward
def drive_forward():
    # Code to control the motors and drive the car forward
    print("Driving forward...")

# Call the function to drive the car forward
drive_forward()
```

Week 2: Selecting the Right Microprocessor for Donkey Car

- Recap and Discussion (10 minutes)
  - Review the homework and discuss the real-world examples of AI in autonomous vehicles.
  - Encourage students to share their findings and insights.

- Introduction to Microprocessors (15 minutes)
  - Explain the role of microprocessors in Donkey Car and their importance in AI applications.
  - Discuss the different types of microprocessors commonly used in DIY robotics.

- Factors to Consider (15 minutes)
  - Discuss the factors to consider when selecting a microprocessor, such as processing power, memory, compatibility, and cost.
  - Explain the trade-offs between different microprocessors and their impact on the performance of Donkey Car.

- Activity: Microprocessor Selection (20 minutes)
  - Provide a list of microprocessors commonly used in Donkey Car projects.
  - Assign students to research and compare the specifications, capabilities, and costs of different microprocessors.
  - Instruct students to select the most suitable microprocessor for their Donkey Car project based on their requirements and budget.

- Recap and Homework (10 minutes)
  - Summarize the key points covered in the lesson.
- Assign homework: Write a short paragraph explaining their microprocessor selection and justification.

Week 2: Selecting the Right Microprocessor for Donkey Car

Python Code Example:
```python
# Example Python code for microprocessor selection

# Define a list of microprocessors
microprocessors = ["Raspberry Pi", "Arduino", "Jetson Nano", "ESP32"]

# Select the most suitable microprocessor based on requirements
selected_microprocessor = "Raspberry Pi"

# Print the selected microprocessor
print("Selected Microprocessor:", selected_microprocessor)
```

MicroPython Code Example:
```python
# Example MicroPython code for microprocessor selection

# Define a list of microprocessors
microprocessors = ["Raspberry Pi", "Arduino", "ESP32"]

# Select the most suitable microprocessor based on requirements
selected_microprocessor = "Raspberry Pi"

# Print the selected microprocessor
print("Selected Microprocessor:", selected_microprocessor)
```

Week 3: Building and Configuring Donkey Car

- Recap and Discussion (10 minutes)
  - Review the homework and discuss the students' microprocessor selections.
  - Encourage students to share their reasoning and discuss any challenges they encountered.

- Building Donkey Car (20 minutes)
  - Provide step-by-step instructions on assembling the Donkey Car components, including the chassis, motor, camera, and microprocessor.
  - Demonstrate the correct wiring and connections.

- Configuring Donkey Car (20 minutes)
  - Explain the process of configuring the microprocessor for Donkey Car.
  - Discuss the necessary software installations, libraries, and dependencies.
  - Guide students through the configuration process, ensuring they understand each step.

- Activity: Test Drive (10 minutes)

  - Allow students to test drive their Donkey Car in a controlled environment.
  - Encourage them to observe and document any issues or errors they encounter.

- Recap and Homework (10 minutes)
  - Summarize the key points covered in the lesson.
- Assign homework: Troubleshoot any issues encountered during the test drive and document the solutions.

Week 3: Building and Configuring Donkey Car

Python Code Example:
```python
# Example Python code for building and configuring Donkey Car

# Import necessary libraries
import RPi.GPIO as GPIO

# Set up GPIO pins for motor control
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(18, GPIO.OUT)

# Function to drive the car forward
def drive_forward():
    GPIO.output(17, GPIO.HIGH)
    GPIO.output(18, GPIO.LOW)

# Call the function to drive the car forward
drive_forward()
```

MicroPython Code Example:
```python
# Example MicroPython code for building and configuring Donkey Car

# Import necessary libraries
from machine import Pin

# Set up GPIO pins for motor control
motor1 = Pin(17, Pin.OUT)
motor2 = Pin(18, Pin.OUT)

# Function to drive the car forward
def drive_forward():
    motor1.on()
    motor2.off()

# Call the function to drive the car forward
```

drive_forward()
```

Week 4: Testing and Implementing Code with Donkey Car

- Recap and Discussion (10 minutes)
  - Review the homework and discuss the solutions to the issues encountered during the test drive.
  - Encourage students to share their experiences and lessons learned.

- Testing Forward Driving (15 minutes)
  - Explain the basics of writing code to control the Donkey Car to drive forward.
  - Demonstrate how to test the code on the Donkey Car.
  - Instruct students to write their own code to make the car drive forward and test it on their Donkey Car.

- Testing Turns (15 minutes)
  - Introduce the concept of turning the Donkey Car left and right.
  - Explain the code modifications required to implement turns.
  - Demonstrate how to test the code on the Donkey Car.
  - Instruct students to modify their code to make the car turn left and right and test it on their Donkey Car.

- Activity: Implementing Remote Control with Autonomy (20 minutes)
  - Explain the concept of remote control with autonomy, where the car can be controlled remotely but also has autonomous capabilities.

- Recap and Homework (10 minutes)
  - Summarize the key points covered in the lesson.
- Assign homework: Write a short paragraph reflecting on their experience testing and implementing code with the Donkey Car.

Week 4: Testing and Implementing Code with Donkey Car

Python Code Example:
```python
# Example Python code for testing and implementing code with Donkey Car

# Import necessary libraries
import time

# Function to drive the car forward
def drive_forward():
    # Code to control the motors and drive the car forward
    print("Driving forward...")

# Function to turn the car left
def turn_left():

```
    # Code to control the motors and turn the car left
    print("Turning left...")

# Call the functions to test the code
drive_forward()
time.sleep(2)
turn_left()
```

MicroPython Code Example:
```python
# Example MicroPython code for testing and implementing code with Donkey Car

# Import necessary libraries
import time

# Function to drive the car forward
def drive_forward():
    # Code to control the motors and drive the car forward
    print("Driving forward...")

# Function to turn the car left
def turn_left():
    # Code to control the motors and turn the car left
    print("Turning left...")

# Call the functions to test the code
drive_forward()
time.sleep(2)
turn_left()
```