

Slip 1:-

Write a program to sense the available networks using Arduino

```
#include <ESP8266WiFi.h>

void setup() {
    Serial.begin(115200);
    delay(1000);

    Serial.println("\nWiFi Network Scanner");
    Serial.println("=====");

    WiFi.mode(WIFI_STA);
    WiFi.disconnect();
    delay(100);
}

void loop() {
    Serial.println("\nScanning for networks...");

    int numNetworks = WiFi.scanNetworks();

    if (numNetworks == 0) {
        Serial.println("No networks found");
    } else {
        Serial.print("Found ");
        Serial.print(numNetworks);
        Serial.println(" networks:");
        Serial.println("-----");

        for (int i = 0; i < numNetworks; i++) {
            Serial.print(i + 1);
            Serial.print(": ");
            Serial.print(WiFi.SSID(i));
            Serial.print(" (");
            Serial.print(WiFi.RSSI(i));
            Serial.print(" dBm)");
            Serial.print(" [");
            Serial.print(getEncryptionType(WiFi.encryptionType(i)));
            Serial.print("]");
            Serial.print(" Channel: ");
            Serial.print(WiFi.channel(i));
            Serial.println();

            delay(10);
        }
    }

    Serial.println("-----");
    delay(10000);
}

String getEncryptionType(wifi_auth_mode_t encryptionType) {
    switch (encryptionType) {
        case WIFI_AUTH_OPEN: return "Open";
        case WIFI_AUTH_WEP: return "WEP";
        case WIFI_AUTH_WPA_PSK: return "WPA-PSK";
        case WIFI_AUTH_WPA2_PSK: return "WPA2-PSK";
        case WIFI_AUTH_WPA_WPA2_PSK: return "WPA/WPA2-PSK";
        case WIFI_AUTH_WPA2_ENTERPRISE: return "WPA2-Enterprise";
    }
}
```

```
        default: return "Unknown";
    }
}
```

Slip 2:-

Write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.

```
const int trigPin = 9;
const int echoPin = 10;
const int ledPin = 13;

void setup() {
    Serial.begin(9600);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    // Clear trig pin
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);

    // Send 10 microsecond pulse
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    // Read echo pin
    duration = pulseIn(echoPin, HIGH);

    // Calculate distance in cm
    distance = duration * 0.034 / 2;

    // Print distance to serial
    Serial.print("Distance: ");
    Serial.print(distance);
    Serial.println(" cm");

    // Control LED based on distance
    if (distance < 50) {
        // Blink fast if object is close
        digitalWrite(ledPin, HIGH);
        delay(100);
        digitalWrite(ledPin, LOW);
        delay(100);
    } else if (distance < 100) {
        // Blink medium speed
        digitalWrite(ledPin, HIGH);
        delay(300);
        digitalWrite(ledPin, LOW);
        delay(300);
    } else {
        // Blink slow if object is far
        digitalWrite(ledPin, HIGH);
        delay(1000);
        digitalWrite(ledPin, LOW);
        delay(1000);
    }
}
```

Slip 3:-

Write a program to detects the vibration of an object with sensor using Arduino.

```
const int vibrationPin = 2;
const int ledPin = 13;
int vibrationState = 0;

void setup() {
    Serial.begin(9600);
    pinMode(vibrationPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    vibrationState = digitalRead(vibrationPin);

    if (vibrationState == HIGH) {
        digitalWrite(ledPin, HIGH);
        Serial.println("Vibration detected!");
        delay(1000);
    } else {
        digitalWrite(ledPin, LOW);
        Serial.println("No vibration");
    }

    delay(100);
}
```

Slip 4:-

Write a program to sense a finger when it is placed on the board Arduino.

```
const int touchPin = 2;
const int ledPin = 13;
int touchState = 0;

void setup() {
    Serial.begin(9600);
    pinMode(touchPin, INPUT);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    touchState = digitalRead(touchPin);

    if (touchState == HIGH) {
        digitalWrite(ledPin, HIGH);
        Serial.println("Finger detected!");
    } else {
        digitalWrite(ledPin, LOW);
        Serial.println("No finger");
    }

    delay(100);
}
```

Slip 5:-

Write a program to connect with the available Wi-Fi using Arduino

```
#include <ESP8266WiFi.h>

const char* ssid = "Your_WiFi_SSID";
const char* password = "Your_WiFi_Password";

void setup() {
    Serial.begin(115200);
    delay(1000);

    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println();
    Serial.println("WiFi connected!");
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());
}

void loop() {
    if (WiFi.status() == WL_CONNECTED) {
        Serial.println("WiFi connected - All good");
    } else {
        Serial.println("WiFi disconnected - Trying to reconnect");
        WiFi.reconnect();
    }

    delay(5000);
}
```

Slip 6:-

Write a program to get temperature notification using Arduino.

```
#include <DHT.h>

#define DHT_PIN 2
#define DHT_TYPE DHT11
#define BUZZER_PIN 8
#define LED_PIN 13

DHT dht(DHT_PIN, DHT_TYPE);
float temperature;

void setup() {
    Serial.begin(9600);
    dht.begin();
    pinMode(BUZZER_PIN, OUTPUT);
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    temperature = dht.readTemperature();

    if (isnan(temperature)) {
        Serial.println("Failed to read temperature!");
        return;
    }

    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.println(" °C");

    if (temperature > 30) {
        Serial.println("ALERT: Temperature too HIGH!");
        digitalWrite(LED_PIN, HIGH);
        tone(BUZZER_PIN, 1000, 1000);
    } else if (temperature < 15) {
        Serial.println("ALERT: Temperature too LOW!");
        digitalWrite(LED_PIN, HIGH);
        tone(BUZZER_PIN, 500, 1000);
    } else {
        Serial.println("Temperature is NORMAL");
        digitalWrite(LED_PIN, LOW);
        noTone(BUZZER_PIN);
    }

    delay(2000);
}
```

Slip 7:-

Write a program for LDR to vary the light intensity of LED using Arduino.

```
const int ldrPin = A0;
const int ledPin = 9;
int ldrValue = 0;
int ledBrightness = 0;

void setup() {
    Serial.begin(9600);
    pinMode(ledPin, OUTPUT);
}

void loop() {
    ldrValue = analogRead(ldrPin);
    ledBrightness = map(ldrValue, 0, 1023, 255, 0);
    analogWrite(ledPin, ledBrightness);

    Serial.print("LDR Value: ");
    Serial.print(ldrValue);
    Serial.print(" | LED Brightness: ");
    Serial.println(ledBrightness);

    delay(100);
}
```

Slip 8:-
Start Raspberry Pi and execute various Linux commands in command terminal window: ls, cd, touch, mv, rm, man, mkdir, rmdir, tar, gzip, cat, more, less, ps, sudo, cron, chown, chgrp, pingetc.

Starting Raspberry Pi
Power up your Raspberry Pi

Login with credentials (default: pi/raspberry)

Open terminal or use SSH to access command line

Basic Linux Commands Examples

bash

1. ls - List directory contents

ls

ls -l # Detailed list

ls -a # Show hidden files

ls -la # Detailed list with hidden files

2. cd - Change directory

cd /home/pi # Change to home directory

cd Documents # Change to Documents folder

cd .. # Go up one directory

cd ~ # Go to home directory

3. touch - Create empty file

touch myfile.txt

touch file1.txt file2.txt file3.txt

4. mv - Move or rename files

mv myfile.txt newfile.txt # Rename

mv file1.txt /home/pi/Documents/ # Move to different directory

5. rm - Remove files

rm myfile.txt

rm -r myfolder/ # Remove directory recursively

rm -f filename # Force remove

6. man - Manual pages

man ls

man cd

man mkdir

7. mkdir - Create directory

mkdir myfolder

mkdir -p parent/child/grandchild # Create nested directories

8. rmdir - Remove empty directory

rmdir emptyfolder

9. tar - Archive files

tar -cvf archive.tar file1.txt file2.txt # Create tar

tar -xvf archive.tar # Extract tar

tar -czvf archive.tar.gz folder/ # Create compressed tar

10. gzip - Compress files

gzip filename.txt # Compress (creates filename.txt.gz)

gzip -d filename.txt.gz # Decompress

```

# 11. cat - Display file content
cat myfile.txt
cat file1.txt file2.txt          # Concatenate multiple files

# 12. more - View file page by page
more longfile.txt
# Press space for next page, q to quit

# 13. less - Better file viewer (can scroll up/down)
less longfile.txt
# Use arrow keys, q to quit

# 14. ps - Process status
ps
ps aux                         # Show all processes
ps -ef | grep python            # Find specific process

# 15. sudo - Execute as superuser
sudo apt update
sudo nano /etc/hosts
sudo reboot

# 16. cron - Schedule tasks
crontab -e                      # Edit cron jobs
# Add: * * * * * /home/pi/myscript.sh  # Run every minute

# 17. chown - Change file owner
sudo chown pi:pi myfile.txt
sudo chown -R pi:pi myfolder/

# 18. chgrp - Change group ownership
sudo chgrp users myfile.txt

# 19. ping - Test network connectivity
ping google.com
ping -c 4 8.8.8.8               # Ping 4 times

# 20. Additional useful commands
pwd                               # Show current directory
whoami                            # Show current user
df -h                             # Disk space
free -h                           # Memory usage
top                               # System monitor

Practical Exercise Session
Create a practice session in your terminal:

bash
# Create a practice directory
mkdir linux_practice
cd linux_practice

# Create some files
touch file1.txt file2.txt file3.txt
ls -l

# Create directories
mkdir folder1 folder2
ls -l

# Move files

```

```
mv file1.txt folder1/
mv file2.txt folder2/

# Check contents
ls folder1/
ls folder2/

# Create a tar archive
tar -cvf myarchive.tar folder1/ folder2/

# Compress it
gzip myarchive.tar

# List files
ls -l

# Remove files and directories
rm file3.txt
rm -r folder1/ folder2/

# Check system info
ps aux | head -10
ping -c 2 google.com

# Clean up
cd ..
rm -r linux_practice/
Useful Tips for Raspberry Pi
bash
# Update system
sudo apt update
sudo apt upgrade

# Check Raspberry Pi info
cat /etc/os-release
vcgencmd measure_temp          # Check CPU temperature

# Check network
ifconfig
iwconfig                      # Wireless info

# Check disk usage
df -h
du -sh /home/pi/              # Check directory size
```

Slip 9 :-

Write python programs on Pi :

- a) Read your name and print Hello message with name
- b) Read two numbers and print their sum, difference, product and division.
- c) Word and character count of a given string.
- d) Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input.

```
# Combined program with menu
import math
```

```
def hello_name():
    name = input("\nEnter your name: ")
    print(f"Hello {name}! Nice to meet you.")
```

```
def calculator():
    num1 = float(input("\nEnter first number: "))
    num2 = float(input("Enter second number: "))

    print(f"\nSum: {num1} + {num2} = {num1 + num2}")
    print(f"Difference: {num1} - {num2} = {num1 - num2}")
    print(f"Product: {num1} × {num2} = {num1 * num2}")

    if num2 != 0:
        print(f"Division: {num1} ÷ {num2} = {num1 / num2:.2f}")
    else:
        print("Division: Cannot divide by zero!")
```

```
def word_character_count():
    text = input("\nEnter a string: ")
    print(f"\nOriginal string: '{text}'")
    print(f"Character count (with spaces): {len(text)}")
    print(f"Character count (without spaces): {len(text.replace(' ', ''))}")
    print(f"Word count: {len(text.split())}")
```

```
def area_calculator():
    print("\n1. Rectangle\n2. Triangle\n3. Circle")
    choice = input("Choose shape (1/2/3): ")

    if choice == '1':
        l = float(input("Enter length: "))
        w = float(input("Enter width: "))
        print(f"Area of rectangle: {l * w:.2f}")

    elif choice == '2':
        b = float(input("Enter base: "))
        h = float(input("Enter height: "))
        print(f"Area of triangle: {0.5 * b * h:.2f}")

    elif choice == '3':
        r = float(input("Enter radius: "))
        print(f"Area of circle: {math.pi * r ** 2:.2f}")

    else:
        print("Invalid choice!")
```

```
# Main menu
while True:
```

```
print("\n" + "="*40)
print("PYTHON PROGRAMS MENU")
print("="*40)
print("1. Hello Name")
print("2. Calculator")
print("3. Word & Character Counter")
print("4. Area Calculator")
print("5. Exit")

choice = input("\nEnter your choice (1-5): ")

if choice == '1':
    hello_name()
elif choice == '2':
    calculator()
elif choice == '3':
    word_character_count()
elif choice == '4':
    area_calculator()
elif choice == '5':
    print("Goodbye!")
    break
else:
    print("Invalid choice! Please try again.")
```

Slip 10:-

Write python programs on Pi like: a) Print a name 'n' times, where name and n are read from standard input, using for and while loops. b) Handle Divided by Zero Exception. c) Print current time for 10 times with an interval of 10 seconds. d) Read a file line by line and print the word count of each line

```
# All in one Python program

import time
from datetime import datetime

def print_name_times():
    print("\n==== Print Name 'n' Times ===")
    name = input("Enter name: ")
    n = int(input("Enter n: "))

    print("Using for loop:")
    for i in range(n):
        print(name)

    print("Using while loop:")
    count = 0
    while count < n:
        print(name)
        count += 1

def handle_division():
    print("\n==== Division with Exception Handling ===")
    try:
        num1 = float(input("Enter numerator: "))
        num2 = float(input("Enter denominator: "))
        result = num1 / num2
        print(f"Result: {result}")
    except ZeroDivisionError:
        print("Error: Cannot divide by zero!")
    except ValueError:
        print("Error: Please enter valid numbers!")

def print_current_time():
    print("\n==== Current Time (10 times with 10s interval) ===")
    for i in range(10):
        current_time = datetime.now().strftime("%H:%M:%S")
        print(f"Time {i+1}: {current_time}")
        time.sleep(10)

def file_word_count():
    print("\n==== File Word Count ===")
    filename = input("Enter filename: ")

    try:
        with open(filename, 'r') as file:
            line_number = 1
            for line in file:
                words = line.split()
                print(f"Line {line_number}: {len(words)} words")
                line_number += 1
    except FileNotFoundError:
        print("Error: File not found!")
    except Exception as e:
```

```
print(f"Error: {e}")

# Main menu
while True:
    print("\n" + "="*50)
    print("PYTHON PROGRAMS MENU")
    print("="*50)
    print("1. Print Name 'n' Times")
    print("2. Division with Exception Handling")
    print("3. Print Current Time")
    print("4. File Word Count")
    print("5. Exit")

    choice = input("\nEnter your choice (1-5): ")

    if choice == '1':
        print_name_times()
    elif choice == '2':
        handle_division()
    elif choice == '3':
        print_current_time()
    elif choice == '4':
        file_word_count()
    elif choice == '5':
        print("Goodbye!")
        break
    else:
        print("Invalid choice! Please try again.")
```