

Code 11

November 19, 2023

```
[ ]: # Calculating average age of patients
def calculate_average_age(ages):
    return sum(ages) / len(ages)

# Counting male and female patients
def count_gender(patients, gender):
    return sum(1 for patient in patients if patient['Gender'] == gender)

# Finding lowest and highest blood pressure readings
def find_blood_pressure_extremes(patients):
    blood_pressures = [tuple(map(int, bp.split('/'))) for bp in
↳patients['BloodPressure']]
    highest_bp = max(blood_pressures)
    lowest_bp = min(blood_pressures)
    return highest_bp, lowest_bp

# Calculating average temperature
def calculate_average_temperature(temperatures):
    return sum(temperatures) / len(temperatures)

# Reading patient data from the file Week13Assignment
file_path = 'Week13Assignment.txt'
with open(file_path, 'r') as file:
    lines = file.readlines()

# Extracting patient information
patients = []
for line in lines:
    name, age, gender, blood_pressure, temperature = line.strip().split(', ')
    patients.append({
        'Name': name,
        'Age': int(age),
        'Gender': gender,
        'BloodPressure': blood_pressure,
        'Temperature': float(temperature)
    })
```

```

# Printing Statistics
average_age = calculate_average_age([patient['Age'] for patient in patients])
male_patients = count_gender(patients, 'Male')
female_patients = count_gender(patients, 'Female')
highest_bp, lowest_bp = find_blood_pressure_extremes(patients)
average_temperature = calculate_average_temperature([patient['Temperature'] for
↪patient in patients])

# Printing Results
print("-- Patient Data Statistics --")
print(f"AverageAge: {average_age:.2f}")
print(f"MalePatients: {male_patients}")
print(f"FemalePatients: {female_patients}")
print(f"HighestBloodPressure: {highest_bp[0]}/{highest_bp[1]}")
print(f"LowestBloodPressure: {lowest_bp[0]}/{lowest_bp[1]}")
print(f"AverageTemperature: {average_temperature:.2f}")

```