Lab 3 Code:

```python
import geopandas as gpd

# Load the GeoJson file into a GeoDataFrame
gdf = gpd.read_file(r"C:\Users\aniqb\OneDrive\Desktop\GIS_SCRIPTS\data.txt")


print(gdf.head())
# view the column names
print(gdf.columns)
# View the shape of the GeoDataFrame
print(gdf.shape)
# view the data type of each column
print(gdf.dtypes)


#Convert the Geographic cordinates to Projected for Area Calculation
gdf = gdf.to_crs("EPSG:26914")

```

```python
class CensusTract:
    """
    A class to represent a census tract.

    Attributes:
    ----------
    geoid : str
        Unique identifier for the census tract.
    population : int
        Population of the census tract.
    geometry : shapely.geometry.polygon.Polygon
        Polygon geometry of the census tract, representing its area.

    Methods:
    -------
    calculate_population_density():
        Calculates and returns the population density of the census tract based on area.
    """
    def __init__(self, geoid, population, geometry):
        """
        Constructs all the necessary attributes for the CensusTract object.

        Parameters:
        ----------
        geoid : str
            Unique identifier for the census tract.
        population : int
            Population of the census tract.
        geometry : shapely.geometry.polygon.Polygon
            Polygon geometry of the census tract, representing its area.
        """
        self.geoid = geoid
        self.population = population
        self.geometry = geometry

```

```python
    def calculate_population_density(self):
        """
        Calculates the population density based on the geometry's area.

        Returns:
        -------
        float
            Population density in people per square kilometer.
        """

        # calculate the population density based on geometry
        ### >>>>>>>>>>>> YOUR CODE HERE <<<<<<<<<<< ###

        # Calculate area in square kilometers
        area_km2 = self.geometry.area / 1e6  # Convert from m² to km²
        # Calculate population density
        population_density = self.population / area_km2
        return population_density

        ### <<<<<<<<<<< END OF YOUR CODE <<<<<<<<<<< ###

```

```python
## Task 2. Calculate the Population Density for Each Census Tract

##Now you have finished the `CensusTract` class, you can use it to calculate the population density for each census tract (each row of the original dataset).

# Function to apply the CensusTract class to each row
def calculate_density(row):
    """
    Applies the CensusTract class to calculate the population density for a given row.

    Parameters:
    ----------
    row : pandas.Series
        A row from the GeoDataFrame containing the census tract information.

    Returns:
    -------
    float
        Calculated population density for the census tract.
    """
    tract = CensusTract(row['GeoId'], row['Pop'], row['geometry'])
    return tract.calculate_population_density()

# Apply the function to each row and add a new column 'Population Density'
gdf['Population Density'] = gdf.apply(calculate_density, axis=1)

# Preview the data with the new column
print(gdf.head())
```

Results:

```
---------Header---------
        GeoId      ALAND  AWATER      Pop_Den   Pop                                           geometry
0  48041000302    4631187       0   901.528877  3954  POLYGON ((-96.39154 30.70355, -96.38845 30.706...
1  48041002023     552593       0  4681.284746  2448  POLYGON ((-96.34722 30.58933, -96.34682 30.589...
2  48041002019    2117268       0  2144.198409  3495  POLYGON ((-96.29598 30.5691, -96.29472 30.5702...
3  48041000207   85875347  352769    44.665008  2775  POLYGON ((-96.54416 30.62831, -96.54349 30.631...
4  48041000104  150913849  487849    30.656749  4523  POLYGON ((-96.45587 30.74766, -96.45576 30.747...
---------Columns---------
Index(['GeoId', 'ALAND', 'AWATER', 'Pop_Den', 'Pop', 'geometry'], dtype='object')
---------Shape---------
(63, 6)
---------Data Types---------
GeoId          int64
ALAND          int32
AWATER         int32
Pop_Den      float64
Pop            int32
geometry    geometry
dtype: object
```

Final Geo-Data frame header after calculating and adding Population Density Column.

```
        GeoId      ALAND  AWATER      Pop_Den   Pop                                           geometry  Population Density
0  48041000302    4631187       0   901.528877  3954  POLYGON ((749828.304 3399653.868, 750117.135 3...          853.128512
1  48041002023     552593       0  4681.284746  2448  POLYGON ((754372.885 3387089.147, 754411.127 3...         4426.478846
2  48041002019    2117268       0  2144.198409  3495  POLYGON ((759342.586 3384962.67, 759459.826 33...         1649.271778
3  48041000207   85875347  352769    44.665008  2775  POLYGON ((735389.082 3390981.951, 735445.466 3...           32.159887
4  48041000104  150913849  487849    30.656749  4523  POLYGON ((743554.031 3404402.657, 743563.421 3...           29.851224
```