Question 1

```
class Employee:
    def __init__(self, employee_id, name, date_of_birth, joining_date, employment_type,
                 base_salary=0, commission=0, deduction=0, cpf_contribution=0,
                 sick_leave=0, sick_pay=0, overtime=0, regular_time=0,
                 address='', phone='', email='', manager=None):
        self.employee_id = employee_id
        self.name = name
        self.date_of_birth = date_of_birth
        self.joining_date = joining_date
        self.employment_type = employment_type
        self.base_salary = base_salary
        self.commission = commission
        self.deduction = deduction
        self.cpf_contribution = cpf_contribution
        self.sick_leave = sick_leave
        self.sick_pay = sick_pay
        self.overtime = overtime
        self.regular_time = regular_time
        self.address = address
        self.phone = phone
        self.email = email
        self.manager = manager

# Sample data
employee1 = Employee(employee_id=1, name="Prateek Tripathi", date_of_birth="1990-05-15",
                     joining_date="2020-01-01", employment_type="full-time",
                     base_salary=5000, cpf_contribution=500, address="123 Main St",
                     phone="123-456-7890", email="prateek@example.com", manager="Aliya")

employee2 = Employee(employee_id=2, name="Pragya Tiwari   ", date_of_birth="1995-08-20",
                     joining_date="2021-03-15", employment_type="part-time",
                     base_salary=2000, cpf_contribution=200, address="456 Elm St",
                     phone="987-654-3210", email="pragya@example.com", manager="Alisha")


# Output sample data in tabular format
print("{:<12} {:<15} {:<12} {:<12} {:<15} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<20} {:<15} {:<20}".format(
    "Employee ID", "Name", "Date of Birth", "Joining Date", "Employment Type",
    "Base Salary", "Commission", "Deduction", "CPF Contribution",
    "Sick Leave", "Sick Pay", "Overtime", "Regular Time", "Address", "Phone", "Email"
))
print("-" * 170)
print("{:<12} {:<15} {:<12} {:<12} {:<15} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<20} {:<15} {:<20}".format(
    employee1.employee_id, employee1.name, employee1.date_of_birth, employee1.joining_date,
    employee1.employment_type, employee1.base_salary, employee1.commission, employee1.deduction,
    employee1.cpf_contribution, employee1.sick_leave, employee1.sick_pay, employee1.overtime,
    employee1.regular_time, employee1.address, employee1.phone, employee1.email
))
print("{:<12} {:<15} {:<12} {:<12} {:<15} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<10} {:<20} {:<15} {:<20}".format(
    employee2.employee_id, employee2.name, employee2.date_of_birth, employee2.joining_date,
    employee2.employment_type, employee2.base_salary, employee2.commission, employee2.deduction,
    employee2.cpf_contribution, employee2.sick_leave, employee2.sick_pay, employee2.overtime,
    employee2.regular_time, employee2.address, employee2.phone, employee2.email
))

```

```
Employee ID  Name            Date of Birth Joining Date Employment Type Base Salary Commission Deduction  CPF Contribution Sick Leave Si
-----------------------------------------------------------------------------------------------------------------------------------
1            Prateek Tripathi 1990-05-15   2020-01-01   full-time       5000       0          0          500        0          0
2            Pragya Tiwari    1995-08-20   2021-03-15   part-time       2000       0          0          200        0          0
```

Question 2

```
from datetime import datetime

class Feedback:
    def __init__(self, name, email, age, gender, location, ratings, recommendations, comments, timestamp, order_id=None):
        self.name = name
        self.email = email
        self.age = age
        self.gender = gender
        self.location = location
        self.ratings = ratings
        self.recommendations = recommendations
        self.comments = comments
        self.timestamp = timestamp
        self.order_id = order_id

    def __str__(self):
        return f"""
        Name: {self.name}
        Email: {self.email}
        Age: {self.age}
        Gender: {self.gender}
        Location: {self.location}
        Ratings: {self.ratings}
        Recommendations: {self.recommendations}
        Comments: {self.comments}
        Timestamp: {self.timestamp}
        Order/Transaction ID: {self.order_id}
        """

class FeedbackForm:
    def __init__(self):
        self.name = input("Name: ")
        self.email = input("Email: ")
        self.age = int(input("Age: "))
        self.gender = input("Gender: ")
        self.location = input("Location: ")
        self.ratings = {}
        self.recommendations = input("Would you recommend us to others? (yes/no): ").lower() == "yes"
        self.comments = input("Any comments or suggestions? (optional): ")
        self.timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        self.order_id = input("Order/Transaction ID (if applicable): ")

        self.ask_ratings()

    def ask_ratings(self):
        print("Please rate the following aspects on a scale of 1 to 5:")
        self.ratings["overall_satisfaction"] = int(input("Overall satisfaction: "))
        self.ratings["product_quality"] = int(input("Product quality: "))
        # Add more rating prompts as needed

    def submit(self):
        feedback = Feedback(
            self.name,
            self.email,
            self.age,
            self.gender,
            self.location,
            self.ratings,
            self.recommendations,
            self.comments,
            self.timestamp,
            self.order_id
        )
        print("Feedback received:")
        print(feedback)

# Sample usage:
feedback_form = FeedbackForm()
feedback_form.submit()

```

```
Name: Prateek
Email: prateek@example.com
Age: 23
Gender: Male
Location: chennai
Would you recommend us to others? (yes/no): yes
Any comments or suggestions? (optional): try to improve sales
```

```
Order/Transaction ID (if applicable): none
Please rate the following aspects on a scale of 1 to 5:
Overall satisfaction: 5
Product quality: 4
Feedback received:

        Name: Prateek
        Email: prateek@example.com
        Age: 23
        Gender: Male
        Location: chennai
        Ratings: {'overall_satisfaction': 5, 'product_quality': 4}
        Recommendations: True
        Comments: try to improve sales
        Timestamp: 2024-04-16 18:44:50
        Order/Transaction ID: none
```

Question 3

```python
class LoanApplication:
    def __init__(self):
        self.demographic_info = {}
        self.education_data = {}
        self.employment_data = {}
        self.prior_loan_data = {}

    def collect_demographic_info(self):
        print("Please provide demographic information:")
        self.demographic_info['name'] = input("Name: ")
        self.demographic_info['age'] = int(input("Age: "))
        self.demographic_info['gender'] = input("Gender: ")
        self.demographic_info['location'] = input("Location: ")

    def collect_education_data(self):
        print("\nPlease provide education data:")
        self.education_data['degree'] = input("Degree: ")
        self.education_data['institution'] = input("Institution: ")
        self.education_data['year_of_completion'] = input("Year of Completion: ")

    def collect_employment_data(self):
        print("\nPlease provide employment data:")
        self.employment_data['employer'] = input("Employer: ")
        self.employment_data['position'] = input("Position: ")
        self.employment_data['income'] = float(input("Monthly Income: "))

    def collect_prior_loan_data(self):
        print("\nPlease provide prior loan data:")
        self.prior_loan_data['loan_amount'] = float(input("Loan Amount: "))
        self.prior_loan_data['loan_purpose'] = input("Loan Purpose: ")
        self.prior_loan_data['repayment_status'] = input("Repayment Status: ")

    def display_application_data(self):
        print("\nLoan Application Data:")
        print("Demographic Information:", self.demographic_info)
        print("Education Data:", self.education_data)
        print("Employment Data:", self.employment_data)
        print("Prior Loan Data:", self.prior_loan_data)

# Collect loan application data
loan_application = LoanApplication()
loan_application.collect_demographic_info()
loan_application.collect_education_data()
loan_application.collect_employment_data()
loan_application.collect_prior_loan_data()

# Display received data
loan_application.display_application_data()

```

```
Please provide demographic information:
Name: Prateek
Age: 23
Gender: Male
Location: Chennai
```

```
Please provide education data:
Degree: B. Tech
Institution: LPU
Year of Completion: 2023

Please provide employment data:
Employer: LV
Position: Analyst
Monthly Income: 45000

Please provide prior loan data:
Loan Amount: 350000
Loan Purpose: Education
Repayment Status: Going on

Loan Application Data:
Demographic Information: {'name': 'Prateek', 'age': 23, 'gender': 'Male', 'location': 'Chennai'}
Education Data: {'degree': 'B. Tech', 'institution': 'LPU', 'year_of_completion': '2023'}
Employment Data: {'employer': 'LV', 'position': 'Analyst', 'income': 45000.0}
Prior Loan Data: {'loan_amount': 350000.0, 'loan_purpose': 'Education', 'repayment_status': 'Going on'}
```