Python Modeling Assignment

Question 1: Bank Account Model with Transactions

```
import random


class BankAccount:
    def __init__(self, account_number, account_type='savings'):
        self.account_number = account_number
        self.account_type = account_type
        self.balance = random.randint(1000, 10000)
        self.transactions = []

    def deposit(self, amount):
        self.balance += amount
        self.transactions.append(f'Deposit: {amount}')

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
            self.transactions.append(f'Withdraw: {amount}')
        else:
            self.transactions.append(f'Withdraw failed: Insufficient balance')

    def generate_random_transactions(self, months, seed_amount):
        for _ in range(random.randint(1, months)):
            if random.choice([True, False]):
```

```
            self.deposit(random.randint(1, seed_amount))
        else:
            self.withdraw(random.randint(1, seed_amount))

    def __repr__(self):
        return f'Account {self.account_number}: Balance = {self.balance}'
```

Output:

Account 24: Balance = 1530

Account 75: Balance = 1980

Account 12: Balance = 2050

...

Account 47: Balance = 9500

Account 9: Balance = 9875

Question 4: Interest Calculation for Bank Account

```
class BankAccountWithInterest:
    def __init__(self, account_number, interest_rate=0.05):
        self.account_number = account_number
        self.balance = random.randint(1000, 10000)
        self.interest_rate = interest_rate
        self.transactions = []

    def withdraw(self, amount):
        if self.balance >= amount:
            self.balance -= amount
        self.transactions.append(('withdraw', amount))
```

```
    def deposit(self, amount):
        self.balance += amount
        self.transactions.append(('deposit', amount))

    def calculate_minimum_balance(self, months):
        min_balances = []
        for _ in range(months):
            min_balance = random.randint(500, self.balance)
            min_balances.append(min_balance)
        return min_balances

    def calculate_interest(self, months):
        min_balances = self.calculate_minimum_balance(months)
        total_min_balance = sum(min_balances)
        interest = (total_min_balance * self.interest_rate) / 12 * months
        return interest

    def __repr__(self):
        return f'Account {self.account_number}: Balance = {self.balance}'
```

Output:

Account 1: Interest for 6 months = 350.75