**GROUP WORK PROJECT** # 3
**Group Number:** 4154

| FULL LEGAL NAME | LOCATION (COUNTRY) | EMAIL ADDRESS | MARK X FOR ANY NON-CONTRIBUTING MEMBER |
|---|---|---|---|
| BIN SUN | Singapore | sunskybin@gmail.com | |
| Jose Roberto Velazquez Gonzalez | Canada | joserobertovg@gmail.com | |
| Jemish Rasikbhai Naliyapara | India | jemish.naliyapara.2508@gmail.com | |

**Statement of integrity:** By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

| | |
|---|---|
| **Team member 1** | **BIN SUN** |
| **Team member 2** | **Jose Roberto Velazquez Gonzalez** |
| **Team member 3** | **Jemish Rasikbhai Naliyapara** |

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
**Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

# Step 1: Fallout Risks

| | Model Failures | Financial Crises |
|---|---|---|
| Scenario 1. Money at a fixed rate for an unsecured purchase (e.g. credit card) for an individual. | Credit scoring models may incorrectly assess creditworthiness, leading to high default rates. | Economic downturns can result in job losses, making it difficult for individuals to repay unsecured credit card debt. |
| Scenario 2. Money at a floating rate for a secured purchase (e.g. home or automobile) for an individual. | Flawed home valuation models could overestimate collateral values, leading to excessive lending against inflated home prices. | Housing market crashes can dramatically reduce home values, turning loans "underwater" and increasing defaults and foreclosures. |
| Scenario 3. Money at a fixed rate for a business for a construction loan. | Inadequate models to forecast construction costs and timelines could lead to cost overruns and project delays, harming loan performance. | Recessions and financial crises could reduce demand, halt projects, and cause defaults on construction loans and assets to lose value. |
| Scenario 4. Publicly traded Equity (e.g. common stock) – that is, securities lending of a stock. | Model inaccuracies may result in investments in underperforming stocks. | Stock market crashes can lead to a rapid and significant loss of investment value. |
| Scenario 5. Publicly traded bond (e.g. treasury bond, corporate bond) – that is, securities lending of a bond. | Bond pricing models such as duration calculations could underestimate interest rate sensitivity, leading to losses from rate changes. | Spikes in risk premiums during crises could significantly drop bond prices and lead to margin calls on leveraged bond positions. |
| Scenario 6. An illiquid security – you choose the security | Limited data could impair valuation models for illiquid assets, leading to mispricing and improper collateral assessments. | Crises reducing market liquidity could make selling or closing positions on illiquid assets extremely challenging. |

## Step 2. Identifying Data

**Scenario 1: Money at a fixed rate for an unsecured purchase (e.g. credit card) for an individual:**

**1. Data Type:** Credit Score, Transaction Data, Payment History, Debt-to-Income Ratios, Personal Information, Credit Limits, Credit Utilization Rates, Customer Inquiries, Interest Rates, etc.

**2. Data Processing:** Daily Transaction Data, Monthly Payment Data, Customer Credit Score Changes, Credit Limit Adjustments, Payment Delinquency Status, Interest Rate Changes, etc.

**3. Data Frequency:** Daily Transaction Data, Monthly Payment Data, Quarterly Credit Score Updates, Real-time Customer Inquiries, etc.

**4. Data Class:** Consumer Credit Data, Payment History Data, Personal Financial Data, Loan/Debt Data, etc.

**5. Data Source:** Credit Bureaus, Transaction Records, Customer Credit Applications, Credit Issuers, Customer Account Statements, etc.

**6. Data Variety:** Historical Credit Score vs. Current Credit Score, Actual Payment vs. Minimum Payment, Credit Utilization vs. Credit Limit, Observed Customer Behavior vs. Modeled Predictions, etc.

**Scenario 2: Money at a floating rate for a secured purchase (e.g. home or automobile) for an individual:**

**1. Data Type:** Mortgage/Loan Data, Property Valuation, Borrower Income, Credit Scores, Interest Rates, Employment Data, Real Estate Market Data, etc.

**2. Data Processing:** Monthly Mortgage Payment Data, Property Appraisal Values, Credit Score Updates, Interest Rate Adjustments, Employment Verification, Real Estate Market Trends, etc.

**3. Data Frequency:** Monthly Mortgage Data, Quarterly Property Valuation Updates, Credit Score Updates, Real-time Interest Rate Adjustments, etc.

**4. Data Class:** Real Estate Data, Mortgage Data, Borrower Financial Data, Credit Data, etc.

**5. Data Source:** Lenders, Property Appraisers, Credit Bureaus, Employment Verification Services, Real Estate Market Data Providers, etc.

**6. Data Variety:** Historical Loan Payment Data vs. Current Payments, Property Valuation vs. Loan Amount, Observed Borrower Behavior vs. Modeled Predictions, etc.

**Scenario 3: Money at a fixed rate for a business for a construction loan:**

1. **Data Type:** Construction Project Data, Financial Statements, Loan-to-Value Ratios, Creditworthiness of the Business, Economic Indicators, Interest Rates, etc.

2. **Data Processing:** Project Milestone Updates, Cash Flow Projections, Loan Repayment History, Economic Forecasts, Loan Interest Rate Adjustments, etc.

3. **Data Frequency:** Project Progress Updates, Quarterly Financial Statements, Business Credit Score Updates, Real-time Economic Indicator Monitoring, etc.

4. **Data Class:** Business Financial Data, Project Specific Data, Credit Data, Economic Data, etc.

5. **Data Source:** Business Financial Reports, Project Management Software, Credit Rating Agencies, Economic Forecasts, Lender's Records, etc.

6. **Data Variety:** Project Progress vs. Projections, Actual Cash Flow vs. Forecasted, Creditworthiness Changes, Observed Economic Trends vs. Modeled Predictions, etc.

**Scenario 4: Publicly traded Equity (e.g. common stock):**

1. **Data Type:** Stock Prices, Earnings Reports, Dividend Yields, P/E Ratios, Market Volatility, Analyst Ratings, News and Events, etc.

2. **Data Processing:** Daily Stock Price Movements, Earnings Surprises, Dividend Changes, Volatility Index Updates, Analyst Recommendations, Sentiment Analysis, etc.

3. **Data Frequency:** High-frequency Stock Price Data, Quarterly Earnings Reports, Dividend Declarations, Daily Market Volatility, Real-time Analyst Updates, etc.

4. **Data Class:** Equity Data, Financial Performance Data, Market Data, Analyst Ratings, etc.

5. **Data Source:** Stock Exchanges, Company Filings, Market Data Providers, Analyst Reports, News Agencies, etc.

6. **Data Variety:** Historical Stock Prices vs. Current Prices, Actual vs. Expected Earnings, Dividend Trends, Observed Market Volatility vs. Modeled Projections, etc.

**Scenario 5: Publicly traded bond (e.g. treasury bond, corporate bond):**

1. **Data Type:** Bond Yields, Credit Ratings, Bond Issuer Financial Statements, Interest Rate Movements, Economic Indicators, Market Sentiment, etc.

2. **Data Processing:** Daily Bond Yield Changes, Credit Rating Updates, Financial Statement Analysis, Interest Rate Trends, Economic Indicator Trends, Sentiment Analysis, etc.

3. **Data Frequency:** Daily Bond Yield Data, Quarterly Credit Rating Updates, Quarterly Financial Statements, Real-time Interest Rate Movements, Economic Indicator Releases, etc.

4. **Data Class:** Fixed Income Data, Credit Data, Economic Data, Market Data, etc.

5. **Data Source:** Bond Market Data Providers, Credit Rating Agencies, Issuer Filings, Economic Indicators, Market News, etc.

6. **Data Variety**: Historical Bond Yields vs. Current Yields, Credit Rating Changes, Observed Interest Rate Trends vs. Modeled Projections, etc.

**Scenario 6: An illiquid security – you choose the security:**

1. **Data Type:** Illiquid Security Transaction Data, Historical Performance, Collateral Valuation, Counterparty Risk, Market Conditions, News and Events, etc.

2. **Data Processing:** Transaction Records, Collateral Valuation Updates, Counterparty Risk Assessment, Real-time Market Condition Monitoring, News Sentiment Analysis, etc.

3. **Data Frequency:** Transaction Data (as available), Collateral Valuation Periodicity, Counterparty Risk Reviews, Real-time Market Condition Updates, News Alerts, etc.

4. **Data Class:** Illiquid Asset Data, Collateral Data, Counterparty Data, Market Data, etc.

5. **Data Source:** OTC Trading Records (if available), Collateral Appraisers, Counterparty Risk Assessment Providers, Market Data Sources, News Agencies, etc.

6. **Data Variety:** Historical Transaction Data vs. Current, Observed Collateral Valuation vs. Modeled, Counterparty Risk Changes, Market Conditions vs. Projections, etc.

The key macroeconomic data would include GDP, unemployment, consumer spending, business investment, etc. to gauge the state of the economy and likelihood of downturns.

Financial market data on asset prices, volatility, risk premia would help assess market crashes and liquidity risks.

Credit ratings and outlook changes from agencies like S&P could provide signals on credit risks and counterparties.

Higher frequency daily or monthly data allows monitoring conditions closely. Economic, fixed income and equity data covers the range of securities.

Comparing projected economic forecasts to actual data can reveal model inaccuracies. Volatility indices versus asset prices can show signals of market instability.

## Step 3. Ethical Considerations based on Fallout

| | Ethical Challenges | Undesirable Results |
|---|---|---|
| Scenario 1. Money at a fixed rate for an unsecured purchase (e.g. credit card) for an individual. | Misleading marketing practices regarding credit offers. Discriminatory lending practices based on demographics. Unfair interest rate increases without clear disclosure. | High consumer debt and financial distress. Disproportionate financial burdens on vulnerable groups. Customer resentment and distrust in financial services. |
| Scenario 2. Money at a floating rate for a secured purchase (e.g. home or automobile) for an individual. | Borrowers exploiting "strategic default" to walk away from underwater mortgages. Lender practices that target financially vulnerable individuals with high-risk loans. Predatory lending practices, misrepresentation of loan terms. | Potential moral hazard, increased foreclosures. Escalating loan defaults and personal financial crises. Exploitative practices, financial hardship for borrowers. |
| Scenario 3. Money at a fixed rate for a business for a construction loan. | Concealing financial risks in project modeling to secure loans. Pressure on businesses to inflate financials for loan approval. Misallocation of construction funds, leading to project abandonment. | Potential business insolvency and loan defaults. Dishonest financial reporting and eventual defaults. Unfinished projects and financial losses. |
| Scenario 4. Publicly traded Equity (e.g. common stock) – that is, securities lending of a<br>stock. | Insider trading and market manipulation. Unethical stock promotion and misinformation. Inaccurate financial reporting, hiding poor company performance. | Market distortions, unfair gains for a select few. Investor losses, damage to market integrity. Investor misjudgment, market volatility, and losses. |
| Scenario 5. Publicly traded bond (e.g. treasury bond, corporate bond) – that is, securities<br>lending of a bond. | Concealing credit risk or misleading bond ratings. Insider trading with access to non-public bond information. Collusion in bond issuance or trading, leading to price | Investor misjudgment, losses, and market instability. Unfair gains for insiders, market manipulation, and losses. Distorted bond market, investor losses, and market |

| | | |
|---|---|---|
| | manipulation. | instability. |
| Scenario 6. An illiquid security – you choose the security | Hiding risks and illiquidity of securities from investors. Speculative practices, creating a market bubble. Overpricing illiquid securities, leading to unfair gains for insiders. | Investor misjudgment, financial losses, and distrust. Asset price bubbles and abrupt market corrections. Unjust enrichment of a select few, market distortions. |

## Step 4. Ethical Considerations based on Data

### Scenario 1: Money at a fixed rate for an unsecured purchase (e.g. credit card) for an individual

| Ethical Challenges | Best Practices |
|---|---|
| Intentionally Creating Misleading Data | Transparency in data collection and reporting. Ensure that data is accurate, and there are safeguards against manipulating credit card loan charge-off rates. |
| Propagating Fake Reviews | Implement mechanisms to verify the authenticity of customer reviews and ratings on financial products and services. Clearly disclose the source of reviews and ratings. |
| Failing to Disclose Fees or Conflicts of Interest | Mandatory disclosure of all fees, interest rates, and potential conflicts of interest in credit card terms and agreements. Even in GWP1 and 2 shows  Fed funds rate variance does not affect the public’s borrowing action. |
| Using Simulated Data Instead of Real Data | Clearly differentiate between simulated data and real data in financial models. Disclose the use of back-testing or simulations. in GWP we use Fred GDP and unemployment real data for analysis which could avoid such challenge |
| Adding Good Data to Bad Data | Collect data from a diverse and representative set of customers to avoid bias. Implement robust data quality checks. No manipulation of th data during GWP 1 and 2 analysis |
| Hiding Data | Avoid selective data omission. Provide complete and accurate information to customers regarding their credit card terms and conditions. |
| Smoothing Data | Report the true volatility and risk associated with credit card loans. Avoid understating risk through data smoothing. in GWP 1 and GWP2 no smoothing function applied for the data analysis |
| Extrapolating Data | Use robust statistical methods and models. Avoid making assumptions that extend beyond the region where data was tested. |

| | |
|---|---|
| Stealing Data | Implement strong data security measures to protect customer information and prevent data breaches. All data for analysis are from trustable public webs |
| Data Quality Problems | Verify and ensure the quality of data used in credit risk assessments and decision-making processes. Use audited financial statements when available. |
| Collecting Data without Permission | Obtain explicit consent from customers before collecting and using their data. Clearly communicate data usage policies in user agreements. |

### Scenario 3: Money at a fixed rate for a business for a construction loan

| Ethical Challenges | Best Practices to Address These Challenges |
|---|---|
| Intentionally Creating Misleading Data | In a construction loan, transparency is key. Accurate and reliable financial data must be provided, and any attempt to manipulate or mislead lenders should be strictly prohibited. Lenders should verify the financial data provided by the borrowers to ensure its accuracy. |
| Propagating Fake Reviews | In the context of a construction loan, it's crucial to have an accurate assessment of the construction project's scope and feasibility. Falsifying project details can lead to adverse consequences. Independent project assessments and due diligence should be carried out to verify the information provided by the borrower. |
| Failing to Disclose Fees or Conflicts of Interest | When obtaining a construction loan, it's vital for all fees and potential conflicts of interest to be fully disclosed. Borrowers and lenders should have a clear understanding of the terms and conditions, including any fees or charges involved. Transparent disclosure is essential. |

| | |
|---|---|
| Using Simulated Data Instead of Real Data | Using simulated or inaccurate data when applying for a construction loan is unethical. Borrowers should provide genuine, historical financial data and project projections without misrepresentation. Lenders should require borrowers to provide audited financial statements to ensure data accuracy. |
| Adding Good Data to Bad Data | In construction lending, providing false or misleading data could lead to incorrect risk assessments. All data provided should be factual, and lenders should independently verify the data's accuracy. |
| Hiding Data | Withholding information in a construction loan application is unacceptable. All relevant data and documentation must be provided to assess the project's viability and potential risks. |
| Smoothing Data | When reporting financial data for a construction loan, transparency is vital. Any attempt to smooth or manipulate data to make the project appear less risky than it is can have severe consequences. |
| Extrapolating Data | Extrapolating data to present an inaccurate picture of the project's future performance is unethical. All projections must be based on reasonable and justifiable assumptions. |
| Stealing Data | Unauthorized access or use of others' data in a construction loan application is illegal and unethical. All data should be obtained and used with the owner's consent and according to applicable laws. |
| Data Quality Problems | In the case of data quality issues in a construction loan application, the best practice is to address and correct these problems before proceeding. Ensuring the quality of financial and project data is essential for risk assessment. |

| | |
|---|---|
| Collecting Data without Permission | When collecting data for a construction loan, borrowers should obtain explicit permission from involved parties and adhere to data privacy laws. |
| Disclosing Data without Permission | Disclosing data without permission in a construction loan application is a violation of privacy and ethics. Borrowers should always obtain proper consent and safeguard data security to prevent unauthorized disclosures. |

### Scenario 4: Publicly traded Equity (e.g. common stock)

| Ethical Challenge | Proposed Best Practices |
|---|---|
| 1. Intentionally Creating Misleading Data | Ensure that the analysis of AMD stock data is based on actual trading data rather than intentionally manipulated or simulated data. Verify the authenticity of the data sources. |
| 2. Propagating Fake Reviews | In stock analysis, rely on credible financial reports, news, and regulatory filings rather than fraudulent reviews. Avoid making investment decisions based on unverified or fake information. |
| 3. Failing to Disclose Fees or Conflicts of Interest | Be transparent about any potential conflicts of interest or fees that might influence stock recommendations. Abide by regulatory requirements for financial advisors to disclose such information. |
| 4. Using Simulated Data Instead of Real Data | Clearly distinguish between simulated data and real transaction data in any analysis. Ensure that back-testing or simulations are not presented as actual historical performance. |
| 5. Adding Good Data to Bad Data | Use accurate and unbiased data when conducting stock analysis. Avoid manipulating data to misrepresent the performance of AMD stock. |

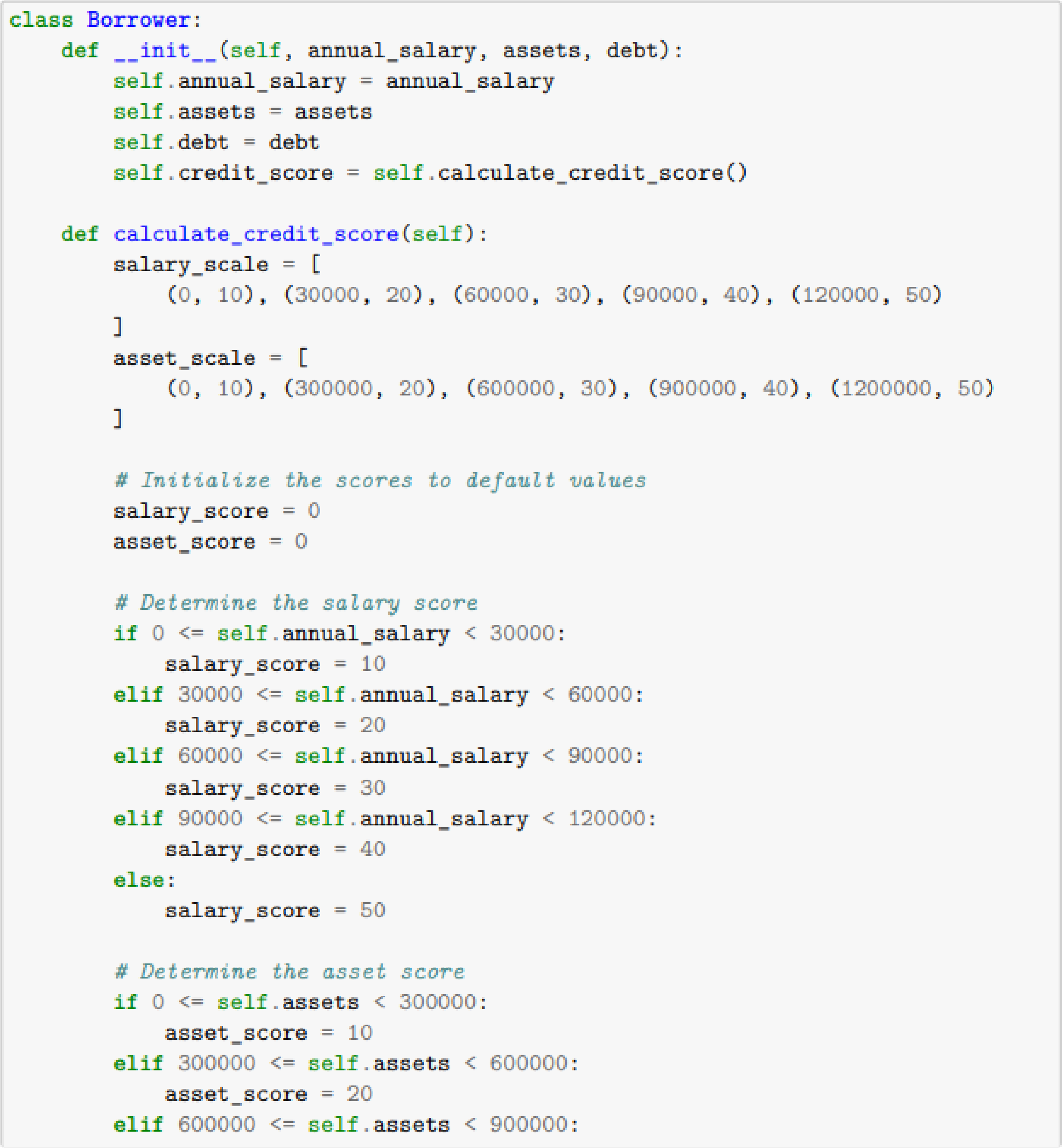
| | |
|---|---|
| | Adhere to industry standards for data quality and reliability. |
| 6. Hiding Data | Provide a comprehensive view of data without selective hiding or removal of information. When analyzing AMD stock, consider all available data, even if it doesn't support a particular narrative. |
| 7. Smoothing Data | Use appropriate data smoothing techniques if they enhance analysis accuracy. Avoid smoothing techniques that underreport volatility, potentially misleading investors. Clearly explain any data adjustments made. |
| 8. Extrapolating Data | When extrapolating trends, ensure that the assumptions and limitations are clearly communicated. Avoid making predictions that significantly deviate from the tested region without proper justification. |
| 9. Stealing Data | Respect data privacy and intellectual property rights. Do not engage in or support data theft for stock analysis. Obtain data through legitimate and legal means. |
| 10. Data Quality Problems | Prioritize data quality in stock analysis. Use audited financial statements and reliable sources to minimize the risk of propagating poor-quality data. Maintain data integrity throughout the analysis process. |
| 11. Collecting Data without Permission | Adhere to data privacy laws and regulations when collecting data for stock analysis. Obtain explicit user consent when necessary and clearly communicate data usage terms. |

| | |
|---|---|
| 12. Disclosing Data without Permission | Safeguard user data and protect it from unauthorized disclosure or breaches. Implement strong data security measures to prevent data leaks and ensure compliance with data protection regulations. |

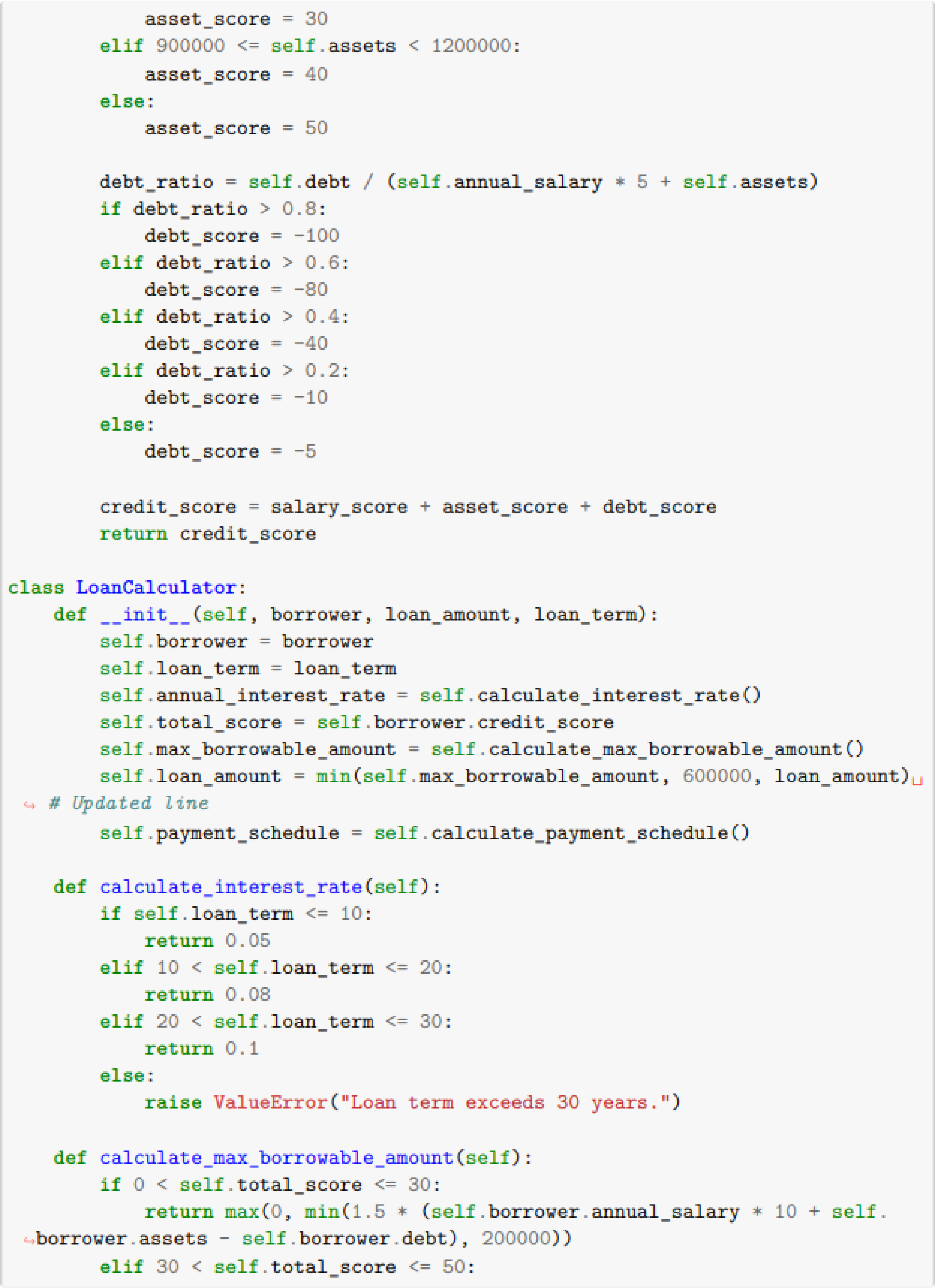
## Step 5. Implementation

### Money Lending Scenarios.

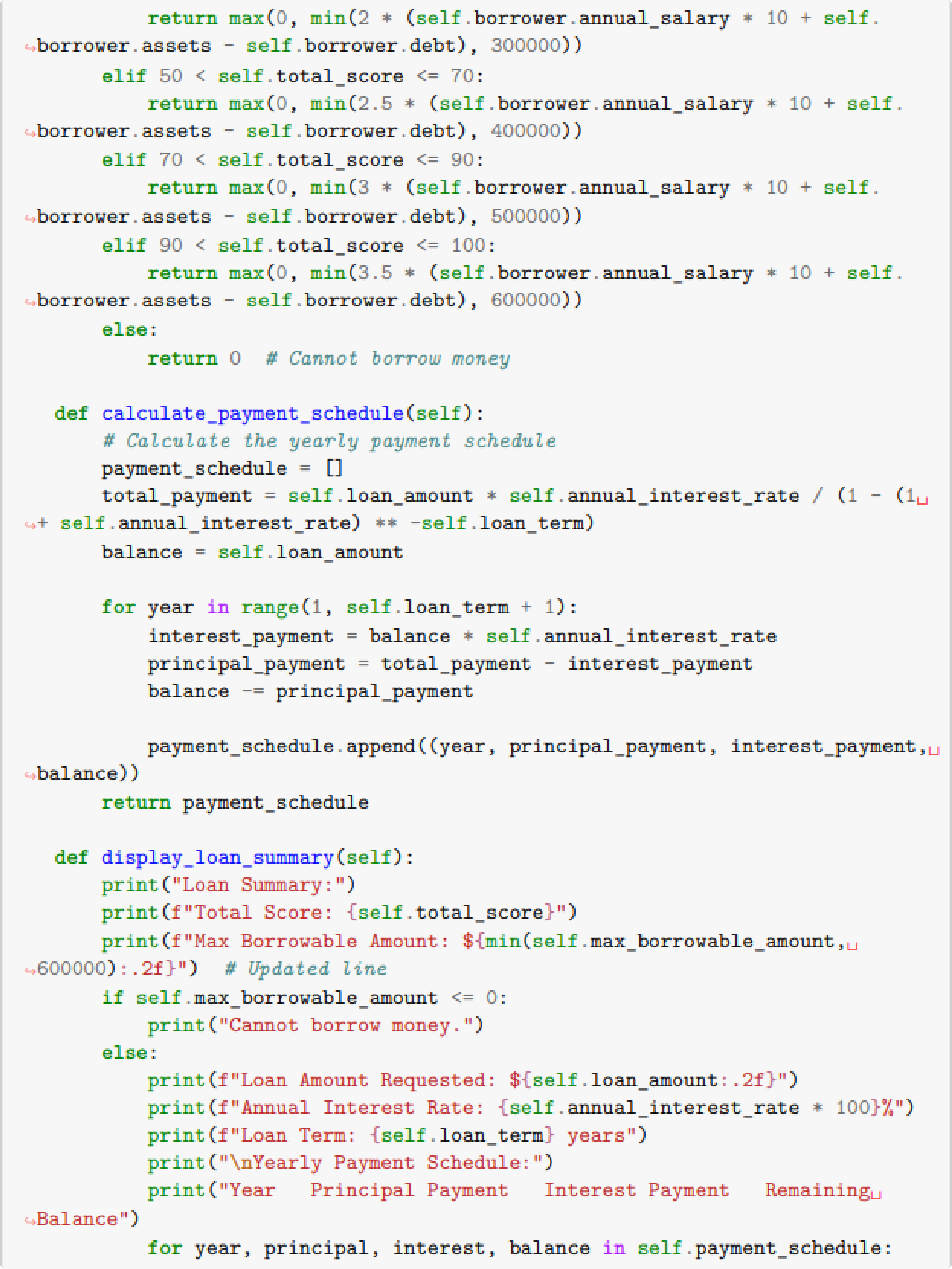
Python code



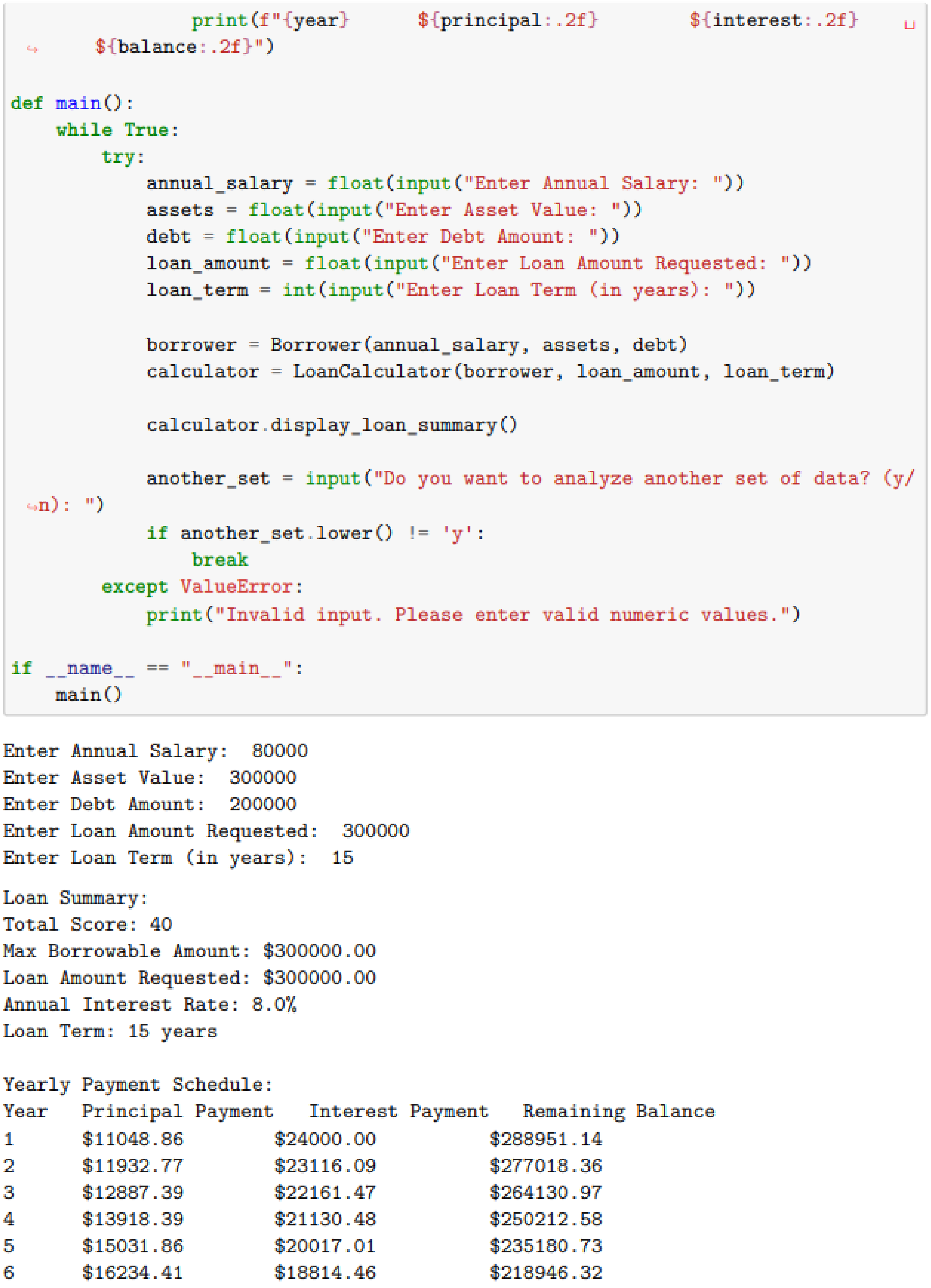
```
class Borrower:
    def __init__(self, annual_salary, assets, debt):
        self.annual_salary = annual_salary
        self.assets = assets
        self.debt = debt
        self.credit_score = self.calculate_credit_score()

    def calculate_credit_score(self):
        salary_scale = [
            (0, 10), (30000, 20), (60000, 30), (90000, 40), (120000, 50)
        ]
        asset_scale = [
            (0, 10), (300000, 20), (600000, 30), (900000, 40), (1200000, 50)
        ]

        # Initialize the scores to default values
        salary_score = 0
        asset_score = 0

        # Determine the salary score
        if 0 <= self.annual_salary < 30000:
            salary_score = 10
        elif 30000 <= self.annual_salary < 60000:
            salary_score = 20
        elif 60000 <= self.annual_salary < 90000:
            salary_score = 30
        elif 90000 <= self.annual_salary < 120000:
            salary_score = 40
        else:
            salary_score = 50

        # Determine the asset score
        if 0 <= self.assets < 300000:
            asset_score = 10
        elif 300000 <= self.assets < 600000:
            asset_score = 20
        elif 600000 <= self.assets < 900000:
```



```
            asset_score = 30
        elif 900000 <= self.assets < 1200000:
            asset_score = 40
        else:
            asset_score = 50

        debt_ratio = self.debt / (self.annual_salary * 5 + self.assets)
        if debt_ratio > 0.8:
            debt_score = -100
        elif debt_ratio > 0.6:
            debt_score = -80
        elif debt_ratio > 0.4:
            debt_score = -40
        elif debt_ratio > 0.2:
            debt_score = -10
        else:
            debt_score = -5

        credit_score = salary_score + asset_score + debt_score
        return credit_score

class LoanCalculator:
    def __init__(self, borrower, loan_amount, loan_term):
        self.borrower = borrower
        self.loan_term = loan_term
        self.annual_interest_rate = self.calculate_interest_rate()
        self.total_score = self.borrower.credit_score
        self.max_borrowable_amount = self.calculate_max_borrowable_amount()
        self.loan_amount = min(self.max_borrowable_amount, 600000, loan_amount)␣
 ↪ # Updated line
        self.payment_schedule = self.calculate_payment_schedule()

    def calculate_interest_rate(self):
        if self.loan_term <= 10:
            return 0.05
        elif 10 < self.loan_term <= 20:
            return 0.08
        elif 20 < self.loan_term <= 30:
            return 0.1
        else:
            raise ValueError("Loan term exceeds 30 years.")

    def calculate_max_borrowable_amount(self):
        if 0 < self.total_score <= 30:
            return max(0, min(1.5 * (self.borrower.annual_salary * 10 + self.
 ↪borrower.assets - self.borrower.debt), 200000))
        elif 30 < self.total_score <= 50:
```

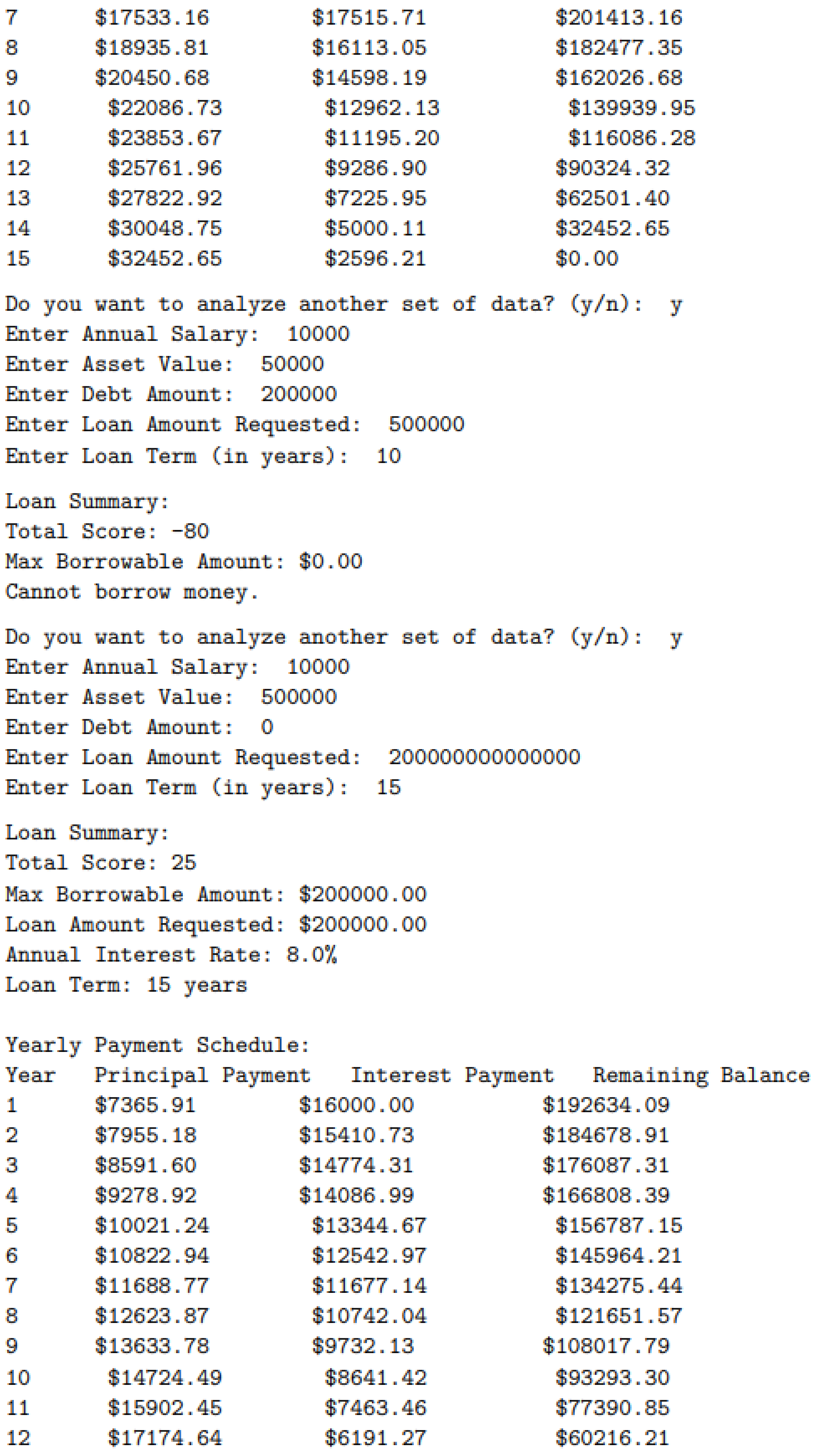


```
            return max(0, min(2 * (self.borrower.annual_salary * 10 + self.
↪borrower.assets - self.borrower.debt), 300000))
        elif 50 < self.total_score <= 70:
            return max(0, min(2.5 * (self.borrower.annual_salary * 10 + self.
↪borrower.assets - self.borrower.debt), 400000))
        elif 70 < self.total_score <= 90:
            return max(0, min(3 * (self.borrower.annual_salary * 10 + self.
↪borrower.assets - self.borrower.debt), 500000))
        elif 90 < self.total_score <= 100:
            return max(0, min(3.5 * (self.borrower.annual_salary * 10 + self.
↪borrower.assets - self.borrower.debt), 600000))
        else:
            return 0  # Cannot borrow money

    def calculate_payment_schedule(self):
        # Calculate the yearly payment schedule
        payment_schedule = []
        total_payment = self.loan_amount * self.annual_interest_rate / (1 - (1␣
↪+ self.annual_interest_rate) ** -self.loan_term)
        balance = self.loan_amount

        for year in range(1, self.loan_term + 1):
            interest_payment = balance * self.annual_interest_rate
            principal_payment = total_payment - interest_payment
            balance -= principal_payment

            payment_schedule.append((year, principal_payment, interest_payment,␣
↪balance))
        return payment_schedule

    def display_loan_summary(self):
        print("Loan Summary:")
        print(f"Total Score: {self.total_score}")
        print(f"Max Borrowable Amount: ${min(self.max_borrowable_amount,␣
↪600000):.2f}")  # Updated line
        if self.max_borrowable_amount <= 0:
            print("Cannot borrow money.")
        else:
            print(f"Loan Amount Requested: ${self.loan_amount:.2f}")
            print(f"Annual Interest Rate: {self.annual_interest_rate * 100}%")
            print(f"Loan Term: {self.loan_term} years")
            print("\nYearly Payment Schedule:")
            print("Year   Principal Payment   Interest Payment   Remaining␣
↪Balance")
            for year, principal, interest, balance in self.payment_schedule:
```

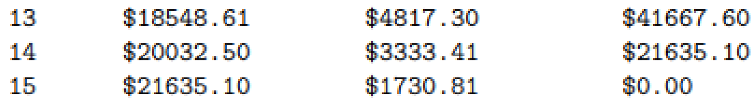


```
                print(f"{year}      ${principal:.2f}       ${interest:.2f}      ${balance:.2f}")

def main():
    while True:
        try:
            annual_salary = float(input("Enter Annual Salary: "))
            assets = float(input("Enter Asset Value: "))
            debt = float(input("Enter Debt Amount: "))
            loan_amount = float(input("Enter Loan Amount Requested: "))
            loan_term = int(input("Enter Loan Term (in years): "))

            borrower = Borrower(annual_salary, assets, debt)
            calculator = LoanCalculator(borrower, loan_amount, loan_term)

            calculator.display_loan_summary()

            another_set = input("Do you want to analyze another set of data? (y/n): ")
            if another_set.lower() != 'y':
                break
        except ValueError:
            print("Invalid input. Please enter valid numeric values.")

if __name__ == "__main__":
    main()
```

```
Enter Annual Salary:  80000
Enter Asset Value:  300000
Enter Debt Amount:  200000
Enter Loan Amount Requested:  300000
Enter Loan Term (in years):  15

Loan Summary:
Total Score: 40
Max Borrowable Amount: $300000.00
Loan Amount Requested: $300000.00
Annual Interest Rate: 8.0%
Loan Term: 15 years

Yearly Payment Schedule:
Year   Principal Payment   Interest Payment   Remaining Balance
1      $11048.86           $24000.00          $288951.14
2      $11932.77           $23116.09          $277018.36
3      $12887.39           $22161.47          $264130.97
4      $13918.39           $21130.48          $250212.58
5      $15031.86           $20017.01          $235180.73
6      $16234.41           $18814.46          $218946.32
```

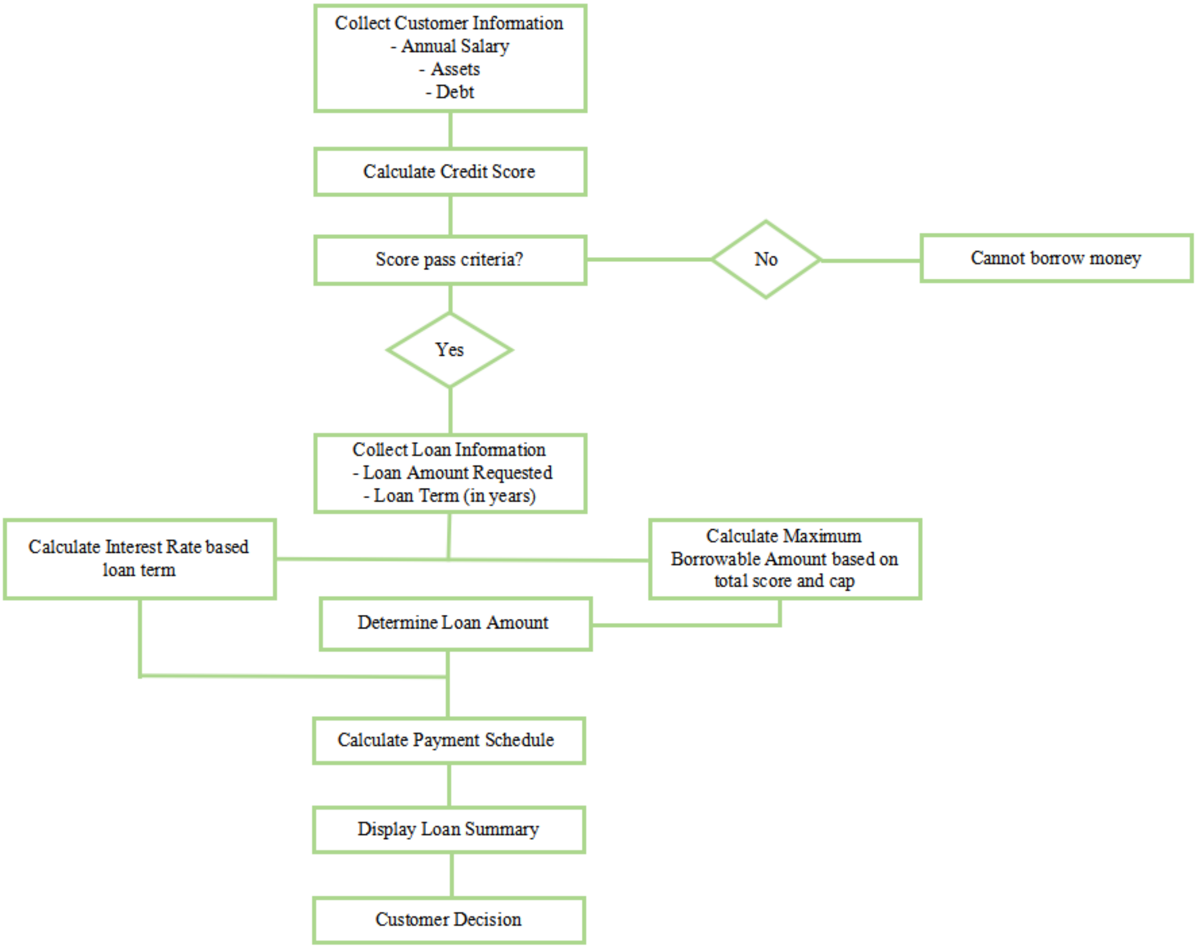


```
7       $17533.16        $17515.71          $201413.16
8       $18935.81        $16113.05          $182477.35
9       $20450.68        $14598.19          $162026.68
10       $22086.73        $12962.13          $139939.95
11       $23853.67        $11195.20          $116086.28
12       $25761.96        $9286.90          $90324.32
13       $27822.92        $7225.95          $62501.40
14       $30048.75        $5000.11          $32452.65
15       $32452.65        $2596.21          $0.00

Do you want to analyze another set of data? (y/n):  y
Enter Annual Salary:  10000
Enter Asset Value:  50000
Enter Debt Amount:  200000
Enter Loan Amount Requested:  500000
Enter Loan Term (in years):  10

Loan Summary:
Total Score: -80
Max Borrowable Amount: $0.00
Cannot borrow money.

Do you want to analyze another set of data? (y/n):  y
Enter Annual Salary:  10000
Enter Asset Value:  500000
Enter Debt Amount:  0
Enter Loan Amount Requested:  200000000000000
Enter Loan Term (in years):  15

Loan Summary:
Total Score: 25
Max Borrowable Amount: $200000.00
Loan Amount Requested: $200000.00
Annual Interest Rate: 8.0%
Loan Term: 15 years


Yearly Payment Schedule:
Year    Principal Payment   Interest Payment   Remaining Balance
1       $7365.91        $16000.00          $192634.09
2       $7955.18        $15410.73          $184678.91
3       $8591.60        $14774.31          $176087.31
4       $9278.92        $14086.99          $166808.39
5       $10021.24        $13344.67          $156787.15
6       $10822.94        $12542.97          $145964.21
7       $11688.77        $11677.14          $134275.44
8       $12623.87        $10742.04          $121651.57
9       $13633.78        $9732.13          $108017.79
10       $14724.49        $8641.42          $93293.30
11       $15902.45        $7463.46          $77390.85
12       $17174.64        $6191.27          $60216.21
```



```
13      $18548.61       $4817.30        $41667.60
14      $20032.50       $3333.41        $21635.10
15      $21635.10       $1730.81        $0.00
```

Workflow diagram



## Summary for Risk Management Team: Loan Eligibility and Risk Assessment System

In this summary, we will explain how a loan eligibility and risk assessment system works, with a focus on the risks involved and how they are managed.

The Borrower Class:

The system begins with the Borrower class, which represents an individual applying for a loan. It takes three main inputs: annual salary, assets, and debt. The borrower's creditworthiness, represented by the credit score, is calculated based on these inputs.

Risks and Management:

Income Risk: The annual salary is a key factor in determining creditworthiness. The credit score may be negatively affected if the borrower has a low salary. This risk is mitigated by using salary scale thresholds, which assign credit scores based on income. Borrowers with higher incomes receive higher credit scores.

Asset Risk: The value of assets also plays a role in credit scoring. If the borrower's assets are low, it may impact their credit score. Like the income risk, asset scale thresholds are used to assign credit scores. Higher asset values result in better credit scores.

Debt Risk: The borrower's existing debt is considered in the credit scoring process. A high debt-to-income ratio can negatively impact the credit score. This risk is managed by assessing the debt ratio and assigning a credit score penalty accordingly.

The LoanCalculator Class:

Once the borrower's credit score is determined, the system calculates the maximum borrowable amount based on this score. It also calculates the annual interest rate for the loan, which depends on the loan term. The borrower's requested loan amount is considered, but it cannot exceed the maximum borrowable amount.

Risks and Management:

Loan Amount Risk: Borrowers may request loan amounts that exceed what they can afford or what they are eligible for based on their credit scores. To mitigate this risk, the system ensures that the loan amount requested is within the limits of the borrower's creditworthiness. This includes a cap of 600,000 USD.

Interest Rate Risk: Longer loan terms come with higher annual interest rates. This risk is managed by setting specific interest rates based on the loan term, with a maximum loan term of 30 years.

Maximum Borrowable Amount Risk: Borrowers with low credit scores might be limited in the amount they can borrow. The system has thresholds based on credit scores, and borrowers can only borrow within these limits.

Risk Management Summary:

The system employs a multi-faceted approach to risk management. Risks related to income, assets, debt, loan amount, and interest rates are systematically assessed and managed. Borrowers are provided with clear information about their eligibility and the associated terms of the loan.

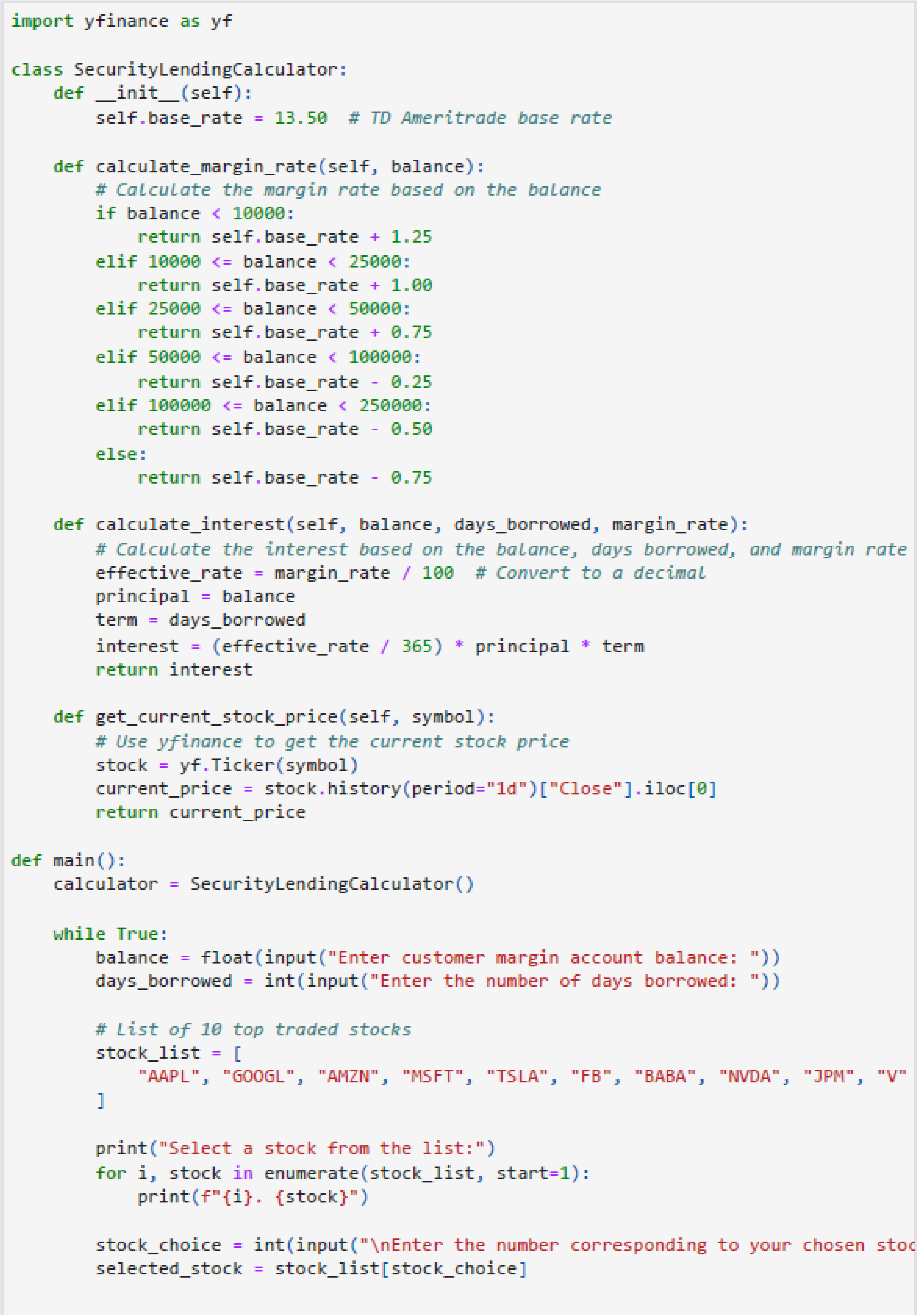
By using thresholds, the system ensures that borrowers are not exposed to loans they cannot reasonably repay. The 600,000 USD cap on loan amounts serves as an additional safeguard. Furthermore, the system's transparency empowers borrowers to make informed decisions and helps the lending institution to maintain a balanced and manageable loan portfolio.

Overall, the system's risk management strategies protect both borrowers and the lending institution from undue financial risks while providing individuals with access to responsible and affordable lending opportunities.

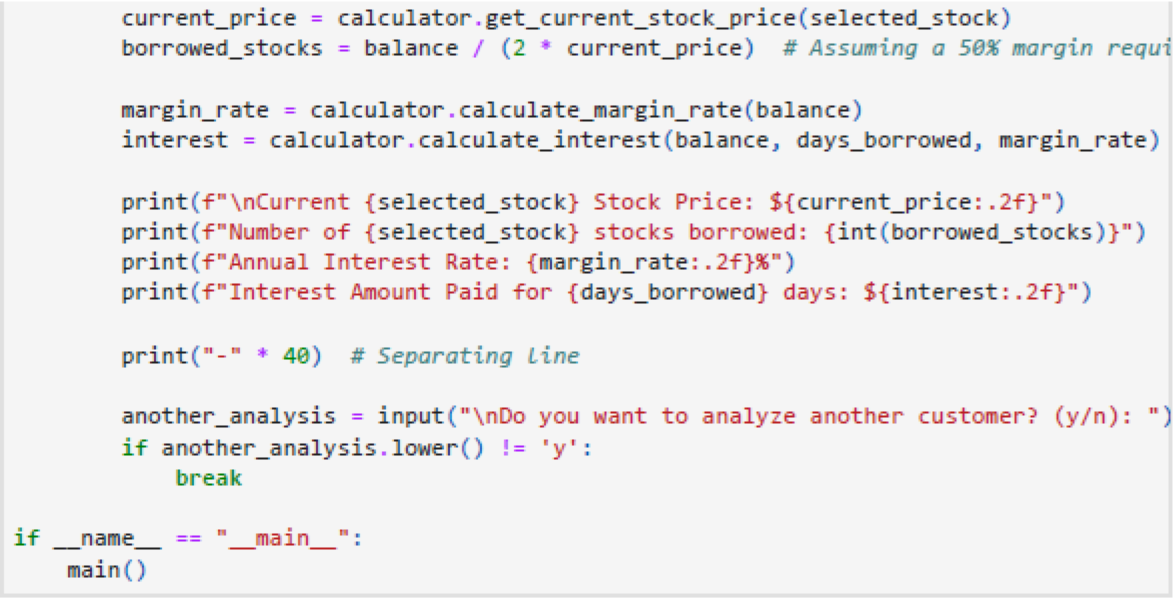
## Step 6.

### Security Lending Scenario

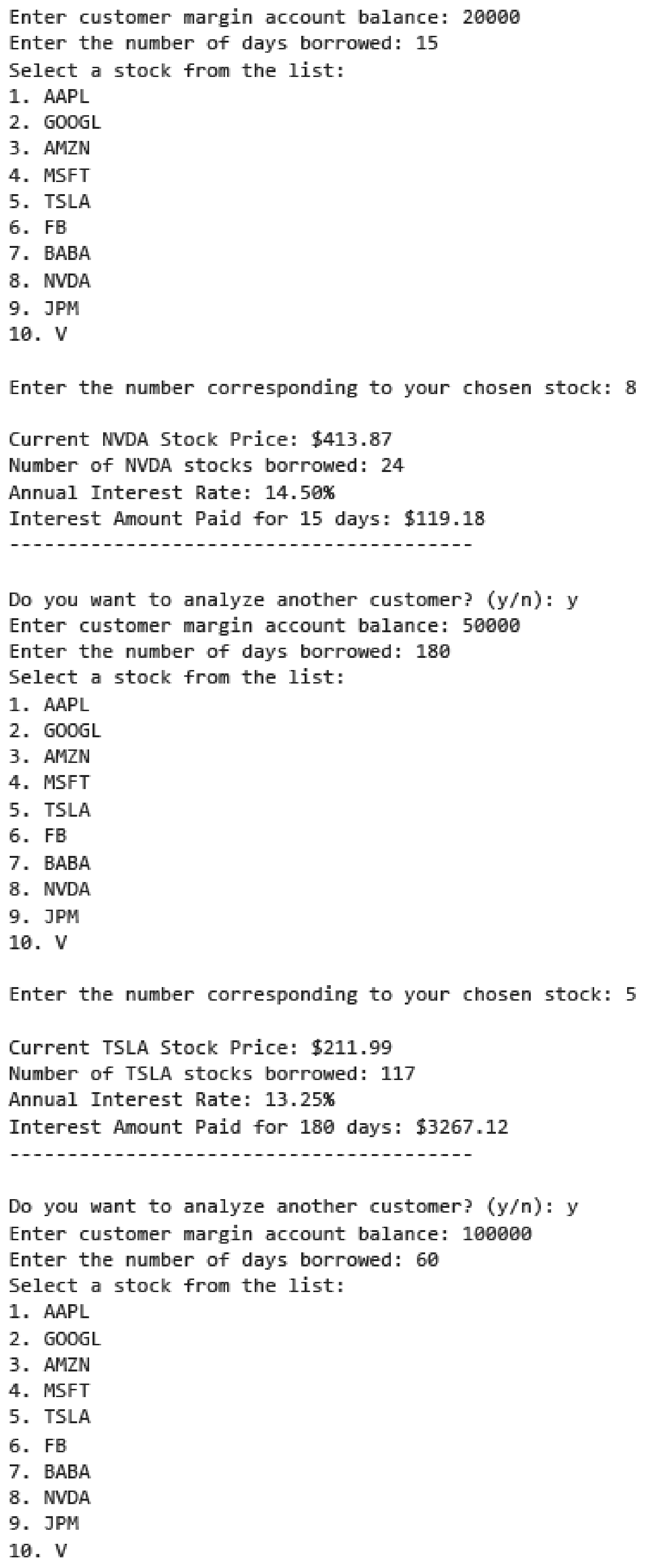
Python Code



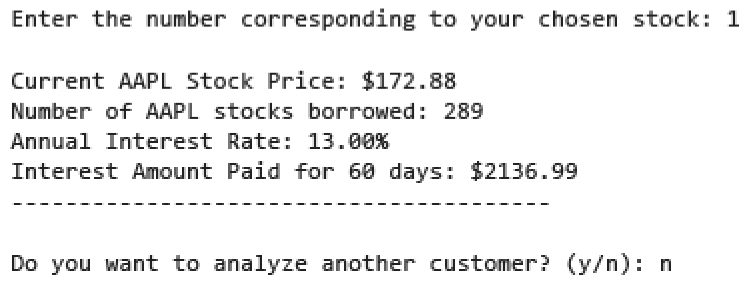
```python
import yfinance as yf

class SecurityLendingCalculator:
    def __init__(self):
        self.base_rate = 13.50  # TD Ameritrade base rate

    def calculate_margin_rate(self, balance):
        # Calculate the margin rate based on the balance
        if balance < 10000:
            return self.base_rate + 1.25
        elif 10000 <= balance < 25000:
            return self.base_rate + 1.00
        elif 25000 <= balance < 50000:
            return self.base_rate + 0.75
        elif 50000 <= balance < 100000:
            return self.base_rate - 0.25
        elif 100000 <= balance < 250000:
            return self.base_rate - 0.50
        else:
            return self.base_rate - 0.75

    def calculate_interest(self, balance, days_borrowed, margin_rate):
        # Calculate the interest based on the balance, days borrowed, and margin rate
        effective_rate = margin_rate / 100  # Convert to a decimal
        principal = balance
        term = days_borrowed
        interest = (effective_rate / 365) * principal * term
        return interest

    def get_current_stock_price(self, symbol):
        # Use yfinance to get the current stock price
        stock = yf.Ticker(symbol)
        current_price = stock.history(period="1d")["Close"].iloc[0]
        return current_price

def main():
    calculator = SecurityLendingCalculator()

    while True:
        balance = float(input("Enter customer margin account balance: "))
        days_borrowed = int(input("Enter the number of days borrowed: "))

        # List of 10 top traded stocks
        stock_list = [
            "AAPL", "GOOGL", "AMZN", "MSFT", "TSLA", "FB", "BABA", "NVDA", "JPM", "V"
        ]

        print("Select a stock from the list:")
        for i, stock in enumerate(stock_list, start=1):
            print(f"{i}. {stock}")

        stock_choice = int(input("\nEnter the number corresponding to your chosen stoc
        selected_stock = stock_list[stock_choice]
```



```
        current_price = calculator.get_current_stock_price(selected_stock)
        borrowed_stocks = balance / (2 * current_price)  # Assuming a 50% margin requi

        margin_rate = calculator.calculate_margin_rate(balance)
        interest = calculator.calculate_interest(balance, days_borrowed, margin_rate)

        print(f"\nCurrent {selected_stock} Stock Price: ${current_price:.2f}")
        print(f"Number of {selected_stock} stocks borrowed: {int(borrowed_stocks)}")
        print(f"Annual Interest Rate: {margin_rate:.2f}%")
        print(f"Interest Amount Paid for {days_borrowed} days: ${interest:.2f}")

        print("-" * 40)  # Separating line

        another_analysis = input("\nDo you want to analyze another customer? (y/n): ")
        if another_analysis.lower() != 'y':
            break

if __name__ == "__main__":
    main()
```

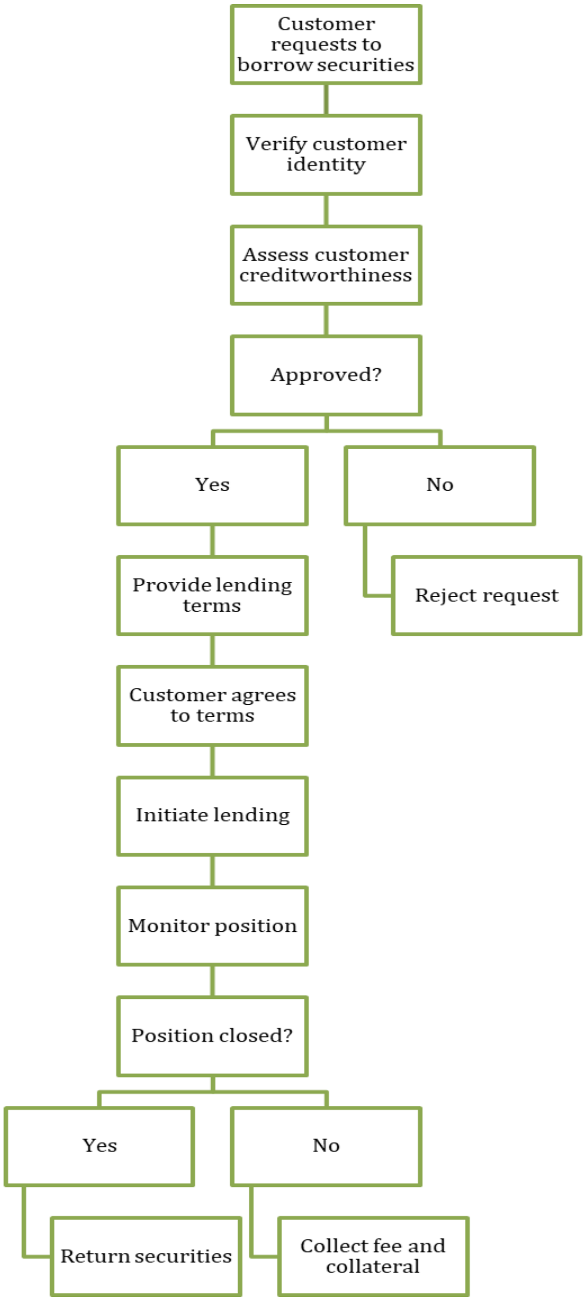


```
Enter customer margin account balance: 20000
Enter the number of days borrowed: 15
Select a stock from the list:
1. AAPL
2. GOOGL
3. AMZN
4. MSFT
5. TSLA
6. FB
7. BABA
8. NVDA
9. JPM
10. V

Enter the number corresponding to your chosen stock: 8

Current NVDA Stock Price: $413.87
Number of NVDA stocks borrowed: 24
Annual Interest Rate: 14.50%
Interest Amount Paid for 15 days: $119.18
----------------------------------------

Do you want to analyze another customer? (y/n): y
Enter customer margin account balance: 50000
Enter the number of days borrowed: 180
Select a stock from the list:
1. AAPL
2. GOOGL
3. AMZN
4. MSFT
5. TSLA
6. FB
7. BABA
8. NVDA
9. JPM
10. V

Enter the number corresponding to your chosen stock: 5

Current TSLA Stock Price: $211.99
Number of TSLA stocks borrowed: 117
Annual Interest Rate: 13.25%
Interest Amount Paid for 180 days: $3267.12
----------------------------------------

Do you want to analyze another customer? (y/n): y
Enter customer margin account balance: 100000
Enter the number of days borrowed: 60
Select a stock from the list:
1. AAPL
2. GOOGL
3. AMZN
4. MSFT
5. TSLA
6. FB
7. BABA
8. NVDA
9. JPM
10. V
```



```
Enter the number corresponding to your chosen stock: 1

Current AAPL Stock Price: $172.88
Number of AAPL stocks borrowed: 289
Annual Interest Rate: 13.00%
Interest Amount Paid for 60 days: $2136.99
----------------------------------------

Do you want to analyze another customer? (y/n): n
```

Workflow diagram



## Summary:

We explained risk management and credit analysis for publicly traded equity to the risk management team in this summary.

We consider the margin rate based on the borrower’s bank balance to mitigate credit risk while developing python code. We only consider 10 highly liquid publicly traded equity in the US stock market. These 10 stocks are as follows:

1. Apple Inc (NASDAQ: AAPL)
2. Alphabet Inc Class A (NASDAQ: GOOGL)
3. Amazon.com, Inc. (NASDAQ: AMZN)
4. Microsoft Corp (NASDAQ: MSFT)
5. Tesla Inc (NASDAQ: TSLA)
6. Meta Platforms Inc (NASDAQ: META)
7. Alibaba Group Holding Ltd-ADR (NYSE: BABA)
8. NVIDIA Corp (NYSE: NVDA)
9. JPMorgan Chase & Co (NYSE: JPM)
10. Visa Inc (NYSE: V)

We calculate the margin based on the borrower’s balance and interest rate. We consider the standard margin as 13.5%. If the borrower’s balance is less than $ 10,000 then the margin is set as 14.75%, as there is a high risk of lending equity with a low balance. While the balance between $10000 and $25000 has a margin of 14.5%. If the borrower has a balance between $25000 and $50000, then the margin is set to 14.25%. We reduce the margin rate as the borrower's account balance increases, such as a balance between $50,000 and $100,000 has a margin rate of 13.25% as it is less risky to lend equity to borrowers with a high balance. We also set the margin rate at 12.75% for borrowers who have a balance greater than $250,000.

We also calculate the interest that is paid by the borrower based on the borrowed equity value and the time period for which the equity was borrowed. This will help us conduct credit risk analysis for each borrower. Additionally, we only chose 10 stocks to lend due to their liquidity. We chose highly liquid stocks so that we can reduce the volatility in return and the market impact. These can also help us to calculate the borrower’s value at risk (VaR). Hence, this Python code dynamically calculates interest based on the borrower’s balance to mitigate the credit risk.

As shown in the flowchart, We first do a creditworthiness analysis of the borrower. If the borrower fulfills the criteria for margin balance, then we initiate the lending of the equity based on the dynamic margin rate and interest calculation based on the period for which the equity was borrowed. We also monitor the borrower’s position, and if it is closed, then the borrower must return the securities, and if it is not closed, then the borrower must pay the fees or collateral. Hence, we can mitigate the credit risk using the dynamic margin rate and interest calculations as implemented in the Python code.

## References

1. Arliss, Will. "Probability of the Challenger Disaster." Medium, 9 October 2020, *https://towardsdatascience.com/probability-of-the-challenger-disaster-c2a17b7984cc.*

2. Murphy, Chris B. "Mortgage Fallout." Investopedia, updated March 23, 2022, *https://www.investopedia.com/terms/m/mortgagefallout.asp.*

3. Liu, Grace. "Data Quality Problems Troubling Business and Financial Researchers: A Literature Review and Synthetic Analysis." Journal of Business & Finance Librarianship, November 2020, *https://digitalcommons.wcupa.edu/lib_facpub/13/.*

4. Huang, Wan and Bai, Yufan. "Customer Identity Concealing and Insider Trading Profitability: Evidence from China" (January 10, 2023). SSRN: *https://ssrn.com/abstract=4321314.*

5. TD Ameritrade. “Margin Rates.” TD Ameritrade, 28 July 2023,
https://www.tdameritrade.com/pricing/margin-and-interest-rates.html Accessed 22 Oct. 2023.