Selected files

7 printable files

src/AmortizedLoan.cpp
src/AmortizedLoan.h
src/Loan.cpp
src/Loan.h
src/main.cpp
src/SimpleLoan.cPP
src/SimpleLoan.h

src/AmortizedLoan.cpp

```
/*
file: AmortizedLoan.cpp
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the implementation file for the AmortizedLoan class.
It contains the implementation of the class functions.

FUNCTIONS:

// Default constructor
AmortizedLoan();
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the AmortizedLoan class.

// Parameterized constructor
AmortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength);
Receives: aPrincipal, aInterestRate, aLoanLength
Returns: Nothing
Description: This is the parameterized constructor for the AmortizedLoan class.

// Destructor
virtual ~AmortizedLoan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the AmortizedLoan class.

// Override the monthly payment function
float monthlyPayment() override;
Receives: Nothing
Returns: a float
Description: This function overrides the monthly payment function from the Loan class.
It calculates the monthly payment for an amortized loan.
*/

#include "AmortizedLoan.h"
#include <cmath>

// Default constructor
AmortizedLoan::AmortizedLoan() {};

```

```
/* ^^^EQUIVALENT TO^^^:
AmortizedLoan::AmortizedLoan() {
    principal = 0.0f;
    interestRate = 0.0f;
    loanLength = 0;
    loanType = "";
}
*/

// Parameterized constructor
AmortizedLoan::AmortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength)
    : Loan(aPrincipal, aInterestRate, aLoanLength, "Amortized Loan") {}

/* ^^^EQUIVALENT TO^^^:
AmortizedLoan::AmortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength) {
    principal = aPrincipal;
    interestRate = aInterestRate;
    loanLength = aLoanLength;
    loanType = "Amortized Loan";
}
*/

// Destructor
AmortizedLoan::~AmortizedLoan() {
    // Empty
}

// Override the monthly payment function
float AmortizedLoan::monthlyPayment() {
    // MONTHLY PAYMENT: (P * R * ((1+R)^L) ) / (((1+R)^L) - 1)
    // P = principal, R = monthly interest rate, L = total months
    float monthlyRate = interestRate / 1200.0f;
    int totalMonths = loanLength * 12;
    float monthlyPayment = (principal * monthlyRate * pow(1 + monthlyRate,
totalMonths)) / (pow(1 + monthlyRate, totalMonths) - 1);
    return monthlyPayment;
}
```

src/AmortizedLoan.h

```
/*
file: AmortizedLoan.h
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the header file for the AmortizedLoan class.
It contains the prototyping for the class.

FUNCTIONS:

// Default constructor
AmortizedLoan();
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the AmortizedLoan class.

```

```
// Parameterized constructor
AmortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength);
Receives: aPrincipal, aInterestRate, aLoanLength
Returns: Nothing
Description: This is the parameterized constructor for the AmortizedLoan class.

// Destructor
virtual ~AmortizedLoan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the AmortizedLoan class.

// Override the monthly payment function
float monthlyPayment() override;
Receives: Nothing
Returns: a float
Description: This function overrides the monthly payment function from the Loan class.
It calculates the monthly payment for an amortized loan.
*/

#ifndef AMORTIZEDLOAN_H
#define AMORTIZEDLOAN_H

#include "Loan.h"

class AmortizedLoan : public Loan {
public:
    // Default constructor
    AmortizedLoan();

    // Parameterized constructor
    AmortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength);

    // Destructor
    virtual ~AmortizedLoan();

    // Override the monthly payment function
    float monthlyPayment() override;
};

#endif
```

src/Loan.cpp

```
/*
file: Loan.cpp
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the implementation file for the Loan class.
It contains the implementation of the class functions.

FUNCTIONS:

// Default constructor
Loan();
```

```
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the Loan class.

// Parameterized constructor
explicit Loan(float aPrincipal, float aInterestRate, int aLoanLength, std::string
aLoanType);
Receives: aPrincipal, aInterestRate, aLoanLength, aLoanType
Returns: Nothing
Description: This is the parameterized constructor for the Loan class.

// Destructor
virtual ~Loan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the Loan class.

// Getters
float getPrincipal();
Receives: Nothing
Returns: a float, principal
Description: This function returns the principal of the loan.

float getInterestRate();
Receives: Nothing
Returns: a float, interestRate
Description: This function returns the annual interest rate of the loan.

float getLoanLength();
Receives: Nothing
Returns: an float, loanLength
Description: This function returns the length of the loan in years.

std::string getLoanType();
Receives: Nothing
Returns: a string, loanType
Description: This function returns the type of the loan.

// Setters
void setPrincipal(float thePrincipal);
Receives: a float, thePrincipal
Returns: Nothing
Description: This function sets the principal of the loan.

void setInterestRate(float theInterestRate);
Receives: a float, theInterestRate
Returns: Nothing
Description: This function sets the annual interest rate of the loan.

void setLoanLength(int theLoanLength);
Receives: an int, theLoanLength
Returns: Nothing
Description: This function sets the length of the loan in years.

void setLoanType(std::string theLoanType);
Receives: a string, theLoanType
```

```
Returns: Nothing
Description: This function sets the type of the loan.

// Base class functions
virtual float monthlyPayment() = 0; // Pure virtual function
Receives: Nothing
Returns: a float, monthlyPayment
Description: This function calculates the monthly payment of the loan.

void displayLoan();
Receives: Nothing
Returns: Nothing
Description: This function displays the loan information to the console.

// File I/O
void saveLoan();
Receives: Nothing
Returns: Nothing
Description: This function saves the loan information to a file.
*/

#include "Loan.h"
#include <iostream>
#include <fstream>
#include <iomanip>
#include <string>

// Default constructor
Loan::Loan() {
    principal = 0.0f;
    interestRate = 0.0f;
    loanLength = 0;
    loanType = "";
}

// Parameterized constructor
Loan::Loan(float aPrincipal, float aInterestRate, int aLoanLength, std::string
aLoanType) {
    principal = aPrincipal;
    interestRate = aInterestRate;
    loanLength = aLoanLength;
    loanType = aLoanType;
}

// Virtual destructor
Loan::~Loan() {
    // Empty
}

// Getters
float Loan::getPrincipal() {
    return principal;
}

float Loan::getInterestRate() {
    return interestRate;
```

```cpp
}

float Loan::getLoanLength() {
    float loanLengthFloat = static_cast<float>(loanLength);
    return loanLengthFloat;
}

std::string Loan::getLoanType() {
    return loanType;
}

// Setters
void Loan::setPrincipal(float thePrincipal) {
    principal = thePrincipal;
}

void Loan::setInterestRate(float theInterestRate) {
    interestRate = theInterestRate;
}

void Loan::setLoanLength(int theLoanLength) {
    loanLength = theLoanLength;
}

void Loan::setLoanType(std::string theLoanType) {
    loanType = theLoanType;
}

// displayLoan: Outputs the loan information to the console
void Loan::displayLoan() {
    std::cout << "Loan Overview" << std::endl;
    std::cout << std::string(32, '=') << std::endl;
    std::cout << std::setw(18) << std::left << "Loan Type:" << getLoanType() << std::endl;
    std::cout << std::setw(18) << std::left << "Principal:" << getPrincipal() << std::endl;
    std::cout << std::setw(18) << std::left << "Interest Rate:" << getInterestRate() << "%" << std::endl;
    std::cout << std::setw(18) << std::left << "Length in Years:" << getLoanLength() << std::endl;
    std::cout << std::setw(18) << std::left << "Monthly Payment:" << monthlyPayment() << std::endl;
}


// saveLoan: Writes the loan information to a file
void Loan::saveLoan() {
    // Directory and file name
    const std::string directory = "data/";
    const std::string filename = "loans.txt";

    // Open file in output mode (overwrites existing file or creates a new one if it doesn't exist)
    std::ofstream outFile(directory + filename, std::ios::out);

    // Write the loan information to the file
    outFile << getPrincipal() << " "
            << getInterestRate() << " "
            << getLoanLength() << " "
            << std::endl;

    // Close the file
    outFile.close();

    /*
    // Check if the file is open
    if (outFile.is_open()) {
        // Write the loan information to the file
        outFile << getPrincipal() << " "
                << getInterestRate() << " "
                << getLoanLength() << " "
                << std::endl;

        // Close the file
        outFile.close();
    } else {
        std::cerr << "Unable to open file for writing." << std::endl;
    }
    */
}

```

src/Loan.h

```
/*
file: Loan.h
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the header file for the Loan class.
It contains the prototyping for the class.

FUNCTIONS:

// Default constructor
Loan();
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the Loan class.

// Parameterized constructor
explicit Loan(float aPrincipal, float aInterestRate, int aLoanLength, std::string aLoanType);
Receives: aPrincipal, aInterestRate, aLoanLength, aLoanType
Returns: Nothing
Description: This is the parameterized constructor for the Loan class.

// Destructor
virtual ~Loan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the Loan class.

// Getters
```

```
float getPrincipal();
Receives: Nothing
Returns: a float, principal
Description: This function returns the principal of the loan.

float getInterestRate();
Receives: Nothing
Returns: a float, interestRate
Description: This function returns the annual interest rate of the loan.

float getLoanLength();
Receives: Nothing
Returns: an float, loanLength
Description: This function returns the length of the loan in years.

std::string getLoanType();
Receives: Nothing
Returns: a string, loanType
Description: This function returns the type of the loan.

// Setters
void setPrincipal(float thePrincipal);
Receives: a float, thePrincipal
Returns: Nothing
Description: This function sets the principal of the loan.

void setInterestRate(float theInterestRate);
Receives: a float, theInterestRate
Returns: Nothing
Description: This function sets the annual interest rate of the loan.

void setLoanLength(int theLoanLength);
Receives: an int, theLoanLength
Returns: Nothing
Description: This function sets the length of the loan in years.

void setLoanType(std::string theLoanType);
Receives: a string, theLoanType
Returns: Nothing
Description: This function sets the type of the loan.

// Base class functions
virtual float monthlyPayment() = 0; // Pure virtual function
Receives: Nothing
Returns: a float, monthlyPayment
Description: This function calculates the monthly payment of the loan.

void displayLoan();
Receives: Nothing
Returns: Nothing
Description: This function displays the loan information to the console.

// File I/O
void saveLoan();
Receives: Nothing
Returns: Nothing
```

```
Description: This function saves the loan information to a file.
*/

#ifndef LOAN_H
#define LOAN_H

#include <string>

class Loan {
protected:
    float principal;
    float interestRate;
    int loanLength;
    std::string loanType;

public:
    // Default constructor
    Loan();

    // Parameterized constructor
    explicit Loan(float aPrincipal, float aInterestRate, int aLoanLength, std::string
aLoanType);

    // Destructor
    virtual ~Loan();

    // Getters
    float getPrincipal();
    float getInterestRate();
    float getLoanLength();
    std::string getLoanType();

    // Setters
    void setPrincipal(float thePrincipal);
    void setInterestRate(float theInterestRate);
    void setLoanLength(int theLoanLength);
    void setLoanType(std::string theLoanType);

    // Base class functions
    virtual float monthlyPayment() = 0; // Pure virtual function
    void displayLoan();

    // File I/O
    void saveLoan();
};

#endif

```

src/main.cpp

```
/*
file: main.cpp
Name: Elijah Heimsoth
Date: 4/11/2024
```

```
Description: This is the main file for the Loan program.
It contains the main function that tests the Loan classes.

FUNCTIONS:

// Main function
int main();
Receives: Nothing
Returns: 0
Description: This is the main function for the Loan program.
*/

#include "Loan.h"
#include "SimpleLoan.h"
#include "AmortizedLoan.h"
#include <iostream>

int main() {
    // Create an empty simple loan
    SimpleLoan emptySimpleLoan;
    std::cout << "Empty Simple Loan:" << std::endl;
    emptySimpleLoan.displayLoan();
    std::cout << std::endl;

    // Create a simple loan with parameters
    // simpleLoan(float aPrincipal, float aInterestRate, int aLoanLength)
    SimpleLoan testSimpleLoan(1000, 5, 4);
    std::cout << "Simple Loan:" << std::endl;
    testSimpleLoan.displayLoan();
    std::cout << std::endl;

    // Create an empty Amortized Loan
    AmortizedLoan emptyAmortizedLoan;
    std::cout << "Empty Amortized Loan:" << std::endl;
    emptyAmortizedLoan.displayLoan();
    std::cout << std::endl;

    // Create an Amortized Loan with parameters
    //amortizedLoan(float aPrincipal, float aInterestRate, int aLoanLength)
    AmortizedLoan testAmortizedLoan(1000, 5, 5);
    std::cout << "Amortized Loan:" << std::endl;
    testAmortizedLoan.displayLoan();
    std::cout << std::endl;

    // Test saveLoan function
    testSimpleLoan.saveLoan();
    testAmortizedLoan.saveLoan();

    return 0;

}
```

src/SimpleLoan.cPP

```
/*
```

```
file: SimpleLoan.cpp
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the implementation file for the SimpleLoan class.
It contains the implementation of the class functions.

FUNCTIONS:

// Default constructor
SimpleLoan();
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the SimpleLoan class.

// Parameterized constructor
explicit SimpleLoan(float aPrincipal, float aInterestRate, int aLoanLength);
Receives: aPrincipal, aInterestRate, aLoanLength
Returns: Nothing
Description: This is the parameterized constructor for the SimpleLoan class.

// Destructor
virtual ~SimpleLoan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the SimpleLoan class.

// Override the monthly payment function
float monthlyPayment() override;
Receives: Nothing
Returns: a float
Description: This function overrides the monthly payment function from the Loan class.
It calculates the monthly payment for a simple loan.
*/

#include "SimpleLoan.h"

// Default constructor
SimpleLoan::SimpleLoan(){};

/* ^^^EQUIVALENT TO^^^:
SimpleLoan::SimpleLoan() {
    principal = 0.0f;
    interestRate = 0.0f;
    loanLength = 0;
    loanType = "";
}
*/

// Parameterized constructor
SimpleLoan::SimpleLoan(float aPrincipal, float aInterestRate, int aLoanLength)
    : Loan(aPrincipal, aInterestRate, aLoanLength, "Simple Loan") {}

/* ^^^EQUIVALENT TO^^^:
SimpleLoan::SimpleLoan(float aPrincipal, float aInterestRate, int aLoanLength) {
    principal = aPrincipal;
    interestRate = aInterestRate;
    loanLength = aLoanLength;
    loanType = "Simple Loan";
}
*/

// Destructor
SimpleLoan::~SimpleLoan() {
    // Empty
}

// Override the monthly payment function
float SimpleLoan::monthlyPayment() {
    // MONTHLY PAYMENT: (P*(R*L + 1)) / L
    // P = principal, R = monthly interest rate, L = total months
    float monthlyRate = interestRate / 1200.0f;
    int totalMonths = loanLength * 12;
    float monthlyPayment = (principal * (monthlyRate * totalMonths + 1)) / totalMonths;
    return monthlyPayment;
}
```

src/SimpleLoan.h

```
/*
file: SimpleLoan.h
Name: Elijah Heimsoth
Date: 4/11/2024
Description: This is the header file for the SimpleLoan class.
It contains the prototyping for the class.

FUNCTIONS:

// Default constructor
SimpleLoan();
Receives: Nothing
Returns: Nothing
Description: This is the default constructor for the SimpleLoan class.

// Parameterized constructor
explicit SimpleLoan(float aPrincipal, float aInterestRate, int aLoanLength);
Receives: aPrincipal, aInterestRate, aLoanLength
```

```
Returns: Nothing
Description: This is the parameterized constructor for the SimpleLoan class.

// Destructor
virtual ~SimpleLoan();
Receives: Nothing
Returns: Nothing
Description: This is the destructor for the SimpleLoan class.

// Override the monthly payment function
float monthlyPayment() override;
Receives: Nothing
Returns: a float
Description: This function overrides the monthly payment function from the Loan class.
It calculates the monthly payment for a simple loan.
*/

#ifndef SIMPLELOAN_H
#define SIMPLELOAN_H

#include "Loan.h"

class SimpleLoan : public Loan {
public:
    // Default constructor
    SimpleLoan();

    // Parameterized constructor
    explicit SimpleLoan(float aPrincipal, float aInterestRate, int aLoanLength);

    // Virtual destructor
    virtual ~SimpleLoan();

    // Override the monthly payment function
    float monthlyPayment() override;
};

#endif
```