SEP101 – Programming Fundamentals
Assignment 1: Real Estate Company
Professor Miguel Watler

Daryl Gonzales
Shay Symonette
Yevhenii (Eugene) Karaman

ASSIGNMENT 1: Real Estate Company

1. **For the apartment, there is a structure of related variables and there are functions which operate on these variables. Write the class declaration for an apartment based on the apartment structure and functions which operate on the apartment structure.**

```
#ifndef _APARTMENT_H_
#define _APARTMENT_H_
#include <string>

class Apartment {
private:
	std::string address;
	int numRooms;
	int numBathrooms;
	double purchasePrice;
	double monthlyRent;
	double monthlyCondoFees;

	double monthlyEarnings;
	double annualROI;
	double currentValue;
	double capitalGains;

	static constexpr double INTEREST_RATE = 0.05;

public:
	// Constructor:
	Apartment();

	// Getter functions:
	std::string GetAddress() const;
	int GetNumRooms() const;
	int GetNumBathrooms() const;
	double GetPurchasePrice() const;
	double GetMonthlyRent() const;
	double GetMonthlyCondoFees() const;

	double GetMonthlyEarnings() const;
	double GetAnnualROI() const;
	double GetCurrentValue() const;
	double GetCapitalGains() const;

	// Parsing helper functions for string validation
	std::string ParseAddress(const std::string& addressStr);
	int ParseNumRooms(const std::string& numRoomsStr);
	int ParseNumBathrooms(const std::string& numBathroomsStr);
	double ParsePurchasePrice(const std::string& purchasePriceStr);
	double ParseMonthlyRent(const std::string& monthlyRentStr);
	double ParseMonthlyCondoFees(const std::string& monthlyCondoFeesStr);

	// Setter functions
	bool SetAddress(const std::string& _address);
	bool SetNumRooms(const int _numRooms);
```

Daryl Gonzales
Shay Symonette
Yevhenii (Eugene) Karaman

```
	bool SetNumBathrooms(const int _numBathrooms);
	bool SetPurchasePrice(const double _purchasePrice);
	bool SetMonthlyRent(const double _monthlyRent);
	bool SetMonthlyCondoFees(const double _monthlyCondoFees);

	// Setter functions that take in the result of the helper functions
	// below
	bool SetMonthlyEarnings(double _monthlyEarnings);
	bool SetAnnualROI(double _annualROI);
	bool SetCurrentValue(double _currentValue);
	bool SetCapitalGains(double _capitalGains);

	// Helper functions
	double CalculateMonthlyEarnings() const;
	double CalculateROI() const;
	double CalculateCurrentValue() const;
	double CalculateCapitalGains() const;
};

#endif // !_APARTMENT_H_
```

2. **For the townhouse, there is a structure of related variables and there are functions which operate on these variables. Write the class declaration for a townhouse based on the townhouse structure and functions which operate on the townhouse structure.**

```
#ifndef _TOWNHOUSE_H_
#define _TOWNHOUSE_H_

#include <string>

class Townhouse {
private:
	std::string address;
	int numRooms;
	int numBathrooms;
	double purchasePrice;
	double monthlyRent;
	double monthlyCondoFees;
	double monthlyUtilities;
	double monthlyPropertyTax;

	double monthlyEarnings;
	double annualROI;
	double currentValue;
	double capitalGains;

	static constexpr double INTEREST_RATE = 0.05;

public:
	// Constructor:
	Townhouse();
```

Daryl Gonzales
Shay Symonette
Yevhenii (Eugene) Karaman

```
        // Getter functions:
        std::string GetAddress() const;
        int GetNumRooms() const;
        int GetNumBathrooms() const;
        double GetPurchasePrice() const;
        double GetMonthlyRent() const;
        double GetMonthlyCondoFees() const;
        double GetMonthlyUtilities() const;
        double GetMonthlyPropertyTax() const;

        double GetMonthlyEarnings() const;
        double GetAnnualROI() const;
        double GetCurrentValue() const;
        double GetCapitalGains() const;

        // Parsing helper functions for string validation
        std::string ParseAddress(const std::string& addressStr);
        int ParseNumRooms(const std::string& numRoomsStr);
        int ParseNumBathrooms(const std::string& numBathroomsStr);
        double ParsePurchasePrice(const std::string& purchasePriceStr);
        double ParseMonthlyRent(const std::string& monthlyRentStr);
        double ParseMonthlyCondoFees(const std::string& monthlyCondoFeesStr);
        double ParseMonthlyUtilities(const std::string& monthlyUtilitiesStr);
        double ParseMonthlyPropertyTax(const std::string& monthlyPropertyTaxStr);

        // Setter functions
        bool SetAddress(const std::string& _address);
        bool SetNumRooms(const int _numRooms);
        bool SetNumBathrooms(const int _numBathrooms);
        bool SetPurchasePrice(const double _purchasePrice);
        bool SetMonthlyRent(const double _monthlyRent);
        bool SetMonthlyCondoFees(const double _monthlyCondoFees);
        bool SetMonthlyUtilities(const double _monthlyUtilities);
        bool SetMonthlyPropertyTax(const double _monthlyPropertyTax);


        // Setter functions that take in the result of the helper functions
        // below
        bool SetMonthlyEarnings(double _monthlyEarnings);
        bool SetAnnualROI(double _annualROI);
        bool SetCurrentValue(double _currentValue);
        bool SetCapitalGains(double _capitalGains);

        // Helper functions
        double CalculateMonthlyEarnings() const;
        double CalculateROI() const;
        double CalculateCurrentValue() const;
        double CalculateCapitalGains() const;
};

#endif // !_TOWNHOUSE_H_
```

Daryl Gonzales
Shay Symonette
Yevhenii (Eugene) Karaman

3. **For the semi-detached house, there is a structure of related variables and there are functions which operate on these variables. Write the class declaration for a semi-detached house based on the semi-detached house structure and functions which operate on the semi-detached house structure.**

```
#ifndef _SEMI_DETACHED_HOUSE_H_
#define _SEMI_DETACHED_HOUSE_H_

#include <string>

class SemiDetachedHouse {
private:
	std::string address;
	int numRooms;
	int numBathrooms;
	double purchasePrice;
	double monthlyRent;
	double monthlyUtilities;
	double monthlyPropertyTax;

	double monthlyEarnings;
	double annualROI;
	double currentValue;
	double capitalGains;

	static constexpr double INTEREST_RATE = 0.05;

public:
	// Constructor:
	SemiDetachedHouse();

	// Getter functions:
	std::string GetAddress() const;
	int GetNumRooms() const;
	int GetNumBathrooms() const;
	double GetPurchasePrice() const;
	double GetMonthlyRent() const;
	double GetMonthlyUtilities() const;
	double GetMonthlyPropertyTax() const;

	double GetMonthlyEarnings() const;
	double GetAnnualROI() const;
	double GetCurrentValue() const;
	double GetCapitalGains() const;

	// Parsing helper functions for string validation
	std::string ParseAddress(const std::string& addressStr);
	int ParseNumRooms(const std::string& numRoomsStr);
	int ParseNumBathrooms(const std::string& numBathroomsStr);
	double ParsePurchasePrice(const std::string& purchasePriceStr);
	double ParseMonthlyRent(const std::string& monthlyRentStr);
	double ParseMonthlyUtilities(const std::string& monthlyUtilitiesStr);
	double ParseMonthlyPropertyTax(const std::string& monthlyPropertyTaxStr);

	// Setter functions
```

Daryl Gonzales
Shay Symonette
Yevhenii (Eugene) Karaman

```
	bool SetAddress(const std::string& _address);
	bool SetNumRooms(const int _numRooms);
	bool SetNumBathrooms(const int _numBathrooms);
	bool SetPurchasePrice(const double _purchasePrice);
	bool SetMonthlyRent(const double _monthlyRent);
	bool SetMonthlyUtilities(const double _monthlyUtilities);
	bool SetMonthlyPropertyTax(const double _monthlyPropertyTax);


	// Setter functions that take in the result of the helper functions
	// below
	bool SetMonthlyEarnings(double _monthlyEarnings);
	bool SetAnnualROI(double _annualROI);
	bool SetCurrentValue(double _currentValue);
	bool SetCapitalGains(double _capitalGains);

	// Helper functions
	double CalculateMonthlyEarnings() const;
	double CalculateROI() const;
	double CalculateCurrentValue() const;
	double CalculateCapitalGains() const;
};

#endif // !_SEMI_DETACHED_HOUSE_H_
```

4. **For the real-estate company, there is a structure of related variables and there are functions which operate on these variables. Write the class declaration for the real-estate company based on the real-estate company structure and functions which operate on the real-estate company structure.**

```
#ifndef _REAL_ESTATE_COMPANY_H_
#define _REAL_ESTATE_COMPANY_H_

// Including some useful libraries and the header files necessary for our code
// We will assume that main() (or a separate header file) handles the File I/O
#include "Apartments.h"
#include "SemiDetachedHouses.h"
#include "Townhouses.h"

class RealEstateCompany {
private:
	const std::string name;
	const std::string address;

	static constexpr int NUM_APARTMENTS = 5;
	static constexpr int NUM_TOWNHOUSES = 5;
	static constexpr int NUM_SEMI_DETACHED_HOUSES = 5;

	Apartment apartment[NUM_APARTMENTS];
	Townhouse townhouse[NUM_TOWNHOUSES];
	SemiDetachedHouse semiDetachedHouse[NUM_SEMI_DETACHED_HOUSES];
public:
	// Constructor with default values
	RealEstateCompany(const std::string& _name = "Brookfield Asset Management",
		const std::string& _address = "11 Yonge Street",
		const int _numApartments = NUM_APARTMENTS,
		const int _numTownhouses = NUM_TOWNHOUSES,
		const int _numSemiDetachedHouses = NUM_SEMI_DETACHED_HOUSES);

	double ApartmentTotalMonthlyEarnings();
	double ApartmentTotalROI();
	double ApartmentAverageROI();
	double ApartmentTotalCurrentValue();
	double ApartmentTotalCapitalGains();

	double TownhouseTotalMonthlyEarnings();
	double TownhouseTotalROI();
	double TownhouseAverageROI();
	double TownhouseTotalCurrentValue();
	double TownhouseTotalCapitalGains();

	double SemiDetachedHouseTotalMonthlyEarnings();
	double SemiDetachedHouseTotalROI();
	double SemiDetachedHouseAverageROI();
	double SemiDetachedHouseTotalCurrentValue();
	double SemiDetachedHouseTotalCapitalGains();

	void PrintReport() const;
};

#endif // !_REAL_ESTATE_COMPANY_H_
```