|  |
| --- |
| **Financial.h** |
| **Line#** | **Code** |
| 1234567891011 | #pragma once#include <string>class Financial {//Utility Classprivate: Financial();public: static double FV(double pv, float r, int n); static double Pmt(double p, float r, int n); static string ToCurrency(double v);}; |





|  |
| --- |
| **Financial.cpp** |
| **Line#** | **Code** |
| 1234567891011121314151617181920212223242526272829303132333435363738394041424344 | #include <iostream>#include <iterator>#include <locale>#include <string>#include <sstream>using namespace std;#include "Financial.h"class MoneyOutputter { locale loc; const money\_put<char>& output; ostringstream os; ostreambuf\_iterator<char, std::char\_traits<char> > iterator;public: MoneyOutputter(const char\* const locale\_name = "US") : loc(locale\_name), output(use\_facet<money\_put<char> >(loc)), iterator(os) { os.imbue(loc); os.setf(ios\_base::showbase); } string toString(double value) { os.str(""); // clear string output.put(iterator, false, os, ' ', value \* 100.0); return os.str(); }};Financial::Financial() {}double Financial::FV(double pv, float r, int n) { return pv \* pow(1.0 + r, n);}double Financial::Pmt(double p, float r, int n) { return p \* ((r \* pow(1 + r, n)) / (pow(1 + r, n) - 1));}string Financial::ToCurrency(double v) { MoneyOutputter outputter; return outputter.toString(v);} |

|  |
| --- |
| **Ex13-03.cpp:** *Class Level Member Functions* |
| **Line#** | **Code** |
| 12345678910111213141516171819202122232425262728293031323334353637 | #include <iostream>using namespace std;#include "Financial.h"#define \_\_DEMO\_\_ 1#if \_\_DEMO\_\_==1int main() { double Saving = 100; float AnnualRate = 0.05; //5% int DurationInYear = 10; //Financial f; cout << "Year\tFV" << endl; for (int year = 0; year <= 10; year++) { cout << year << '\t' << Financial::ToCurrency(Financial::FV(Saving, AnnualRate, year)) << endl; } return 0;}#elseint main() { double HousePrice = 1000000; float LoanPercentage = 0.9F; double Loan = HousePrice \* LoanPercentage; float AnnualRate = 0.046F; int DurationInYear = 30; double MonthlyInstallment = Financial::Pmt(Loan, AnnualRate / 12, DurationInYear \* 12); cout << "Monthly Installment:" << Financial::ToCurrency(MonthlyInstallment) << endl; cout << "Min Net Income:" << Financial::ToCurrency(3 \* MonthlyInstallment) << endl; cout << "Total Payment:" << Financial::ToCurrency(MonthlyInstallment \* DurationInYear \* 12) << endl; return 0;}#endif |