**CPP Problem Design**

|  |
| --- |
| **Subject:** **Bank** |
| **Contributor: 易可鈞** |
| **Main testing concept:** Class   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  □ FLOW OF CONTROL  ■ FUNCTION BASICS  □ PARAMETERS AND OVERLOADING  □ ARRAYS  ■ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  □ OPERATOR OVERLOADING, FRIENDS,AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description:**  Define a class for a type called Bank. Include a function that allows the user to make deposit into their savings account(bank.deposit), a function that allows the user to get loans from bank(bank.loan), a function that allows the user to check their balance in their savings account(bank.getBalance()), and a function that allows the user to know how much they owe the bank(bank.getDebt()). Also include a function that initiates the repayment process(bank.payMonthlyBill()).  You should write you class under the given ”Bank.cpp”, as we will use our own main.cpp to call your class function. And you should NOT modify “Bank.h”, as the header file will be reverted back to its original state when you submit your code.  When the program starts, a creditscore of 200 should be initialized.  When the user tries to get loans from the bank, the creditscore will be used to decide how much money the user can borrow at most.  If 150<creditscore<=200, user can borrow at most 5 times of their balance in savings account.  If 100<creditscore<=150, user can borrow at most 3 times of their balance in savings account.  If 50<creditscore<=100, user can borrow at most 2 times of their balance in savings account.  If 20<creditscore<=50, user can borrow at most 1 times of their balance in savings account.  If the creditscore<=20, The bank will not lend any money.  If user request more than they can borrow, they will get the maximum allowed loan from the bank.  In the repayment function, some, or all of the user’s deposits will be used to repay the debt. Both the status of the balance and debt will be used to determine the amount of repayment. The creditscore will also change depending on how much the user can repay the debt. If the balance can pay off the debt, the user pay off the debt with the balance and the creditscore will increase by 50. If the balance cannot pay off the debt, but is sufficient to cover half of it, the user repay half of the debt with the balance and the creditscore will increase by 20. If the balance is less than half of the debt, but is sufficient to cover one-tenth of the debt, the user repay one-tenth of the debt with the balance and the creditscore will decrease by 30. If the balance is less than one-tenth of the debt, all of the balance will be used to repay the debt and the creditscore will decrease by 100. Please beware that the maximum of the creditscore is 200, and the minimum of the creditscore is 0. If the debt is not fully paid up after the repayment, the rest of the debt will be multiplied by an interest rate of 1.01. Furthermore, a late-fee of $300 will be charged(added to the debt) after the multiplication if less than one-tenth of the debt is repaid.  **Input:**  No input.  **Output:**  As following sample.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | int main()  {  Bank bank = Bank();  bank.deposit(100);  bank.loan(150);    printf("Deposit Made: 100 Requested Loan:150\n");  printf("Check balance: %.3f ", bank.getBalance());  printf("Check debt: %.3f\n", bank.getDebt());  bank.payMonthlyBill();  printf("New balance: %.3f ", bank.getBalance());  printf("New debt: %.3f\n\n", bank.getDebt());  bank.deposit(100);  bank.loan(1500);  printf("Deposit Made: 100 Requested Loan:1500\n");  printf("Check balance: %.3f ", bank.getBalance());  printf("Check debt: %.3f\n", bank.getDebt());  bank.payMonthlyBill();  printf("New balance: %.3f ", bank.getBalance());  printf("New debt: %.3f\n\n", bank.getDebt());  bank.deposit(500);  bank.loan(2000);  printf("Deposit Made: 500 Requested Loan:2000\n");  printf("Check balance: %.3f ", bank.getBalance());  printf("Check debt: %.3f\n", bank.getDebt());  bank.payMonthlyBill();  printf("New balance: %.3f ", bank.getBalance());  printf("New debt: %.3f\n\n", bank.getDebt());  return 0;  } | Deposit Made: 100 Requested Loan:150  Check balance: 100.000 Check debt: 150.000  New balance: 25.000 New debt: 75.750  Deposit Made: 100 Requested Loan:1500  Check balance: 125.000 Check debt: 700.750  New balance: 54.925 New debt: 636.982  Deposit Made: 500 Requested Loan:2000  Check balance: 554.925 Check debt: 2636.982  New balance: 291.227 New debt: 2397.016 | |  |  | |
| **■ Easy,Only basic programming syntax and structure are required.**  **□ Medium,Multiple programming grammars and structures are required.**  **□ Hard,Need to use multiple program structures or more complex data types.** |
| **Expected solving time:**  50 minutes |
| **Other notes:** |