Computational Problem Solving II

CPET-321

Lab #4

RIT | College of Engineering Technology

Assignments: Saturday 10/1 @ 11:59 PM

- zyBook

Lab #4 60 pts

Due: 10/01/2022, 11:59 PM EDT

Shown to students

18.12 LAB: Date (cin & cout Date ADT) (GHZ - 10)	20 pts
18.13 LAB: Date (Class Conversion Date ADT -> Long Int) (GHZ - 10)	20 pts
18.14 LAB: Date (Class Conversion Long int -> Date ADT) (GHZ - 10)	20 pts

- Visual Studio

BankAccount

- *main.cpp*
- *BankAccount.cpp*
- *BankAccount.h*

zyBook Code

18.12 LAB: Date #1 (cin & cout for Date)

Given main() and the outline for the declarations (date1.h) and definitions (date1.cpp) for the class **Date**, add input/output capability so that the standard **cin** and **cout** functions work for the ADT class Date.

Files:

- main.cpp..... Complete
- date1.h..... Partial, need to complete
- date1.cpp..... Partial, need to complete

To add **cin** capability...

- Create an overloaded operator function for the insertion (>>) operator for the istream.
- Give this overloaded operator function (i.e. >>) access to the ADT's data-members by means of a friend function.

To add **cout** capability...

- Create an overloaded operator function for the extraction (<<) operator for the ostream.
- Give this overloaded operator function (i.e <<) access to the ADT's data-members by means of a friend function.

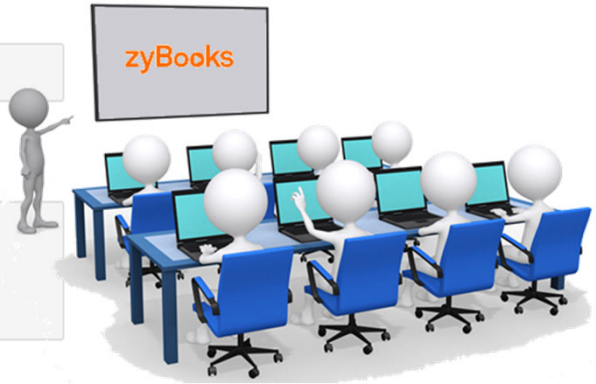
No changes should be made to main().

Ex. If the input is:

```
12 15 2014
```

The output should be:

```
Date A = 12/18/2012
Data B = 09/25/2017
Data C = 12/15/2014
```



18.13 LAB: Date #2 (Class Conversion)

Given main() and the outline for the declarations (date2.h) and definitions (date2.cpp) for the class **Date**, add a **Conversion Operator Function** that converts an object of class Date to a long integer using the following format:

Date	-->	long int
MM/DD/YYYY	-->	YYYYMMDD

Files:

- main.cpp..... Complete
- date2.h..... Partial, need to complete
- date2.cpp..... Partial, need to complete

For Example:

Date	-->	long int
12/03/2014	-->	20141203

Note: This lab builds on the results of **Lab: Date #1**. You will need to copy the code from date1.h and date1.cpp to this solution.

No changes should be made to main().

Ex. If the input is:

```
12 3 2014
```

The output should be:

```
Date A = 12/18/2012
Data A (num) = 20121218
Data B = 09/25/2017
Data B (num) = 20170925
Data C = 12/03/2014
Data C (num) = 20141203
```

18.14 LAB: Date #3 (Class Conversion)

Given main() and the outline for the declarations (date3.h) and definitions (date3.cpp) for the class **Date**, add a **Constructor Function** that converts a long integer into an object of class Date using the following format:

long int	-->	Date
YYYYMMDD	-->	MM/DD/YYYY

Files:

- main.cpp..... Complete
- date3.h..... Partial, need to complete
- date3.cpp..... Partial, need to complete

For Example:

long int	-->	Date
20141203	-->	12/03/2014

Note: This lab builds on the results of **Lab: Date #2**. You will need to copy the code from date2.h and date2.cpp to this solution.

No changes should be made to main().

Ex. If the input is:

```
20141203
```

The output should be:

```
Date A = 12/18/2012
Data A (num) = 20121218
Data B (num) = 20141203
Date B = 12/03/2014
```

VISUAL STUDIO CODE

Bank Accounts – Specifications (1 of 5)

Given main() (mostly), code the class **BankAccount** (in files `BankAccount.h` and `BankAccount.cpp`) that manage an individuals checking and savings accounts.

Bank Accounts – Specifications (2 of 5)

The **BankAccount** class contains the following data members:

- customer name (string)
- savings account balance (double)
- checking account balance (double)

Bank Accounts – Specifications (3 of 5)

The **BankAccount** class contains the following constructor and member functions:

- BankAccount(string, double, double);
  - The 1st parameter is the account owners name
  - The 2nd parameter is the initial balance of the checking account
  - The 3rd parameter is the initial balance of the savings account
- void DepositChecking(double);
  - The parameter is the amount added to the checking account balance.
- void DepositSavings(double)
  - The parameter is the amount added to the savings account balance.

Note: All amounts should be positive, if not, the about should not be added/subtracted from the account

Bank Accounts – Specifications (4 of 5)

The **BankAccount** class contains the following constructor and member functions:

- void WithdrawChecking(double);
  - The parameter is the amount subtracted from the checking account balance.
- void WithdrawSavings(double);
  - The parameter is the amount subtracted from the savings account balance.
- void TransferToChecking(double);
  - The parameter is the amount subtracted from the savings account balance and add to the checking account balance.
- void TransferToSavings(double);
  - The parameter is the amount subtracted from the checking account balance and add to the savings account balance.

Note: All amounts should be positive, if not, the about should not be added/subtracted from the account

Bank Accounts – Specifications (5 of 5)

The **BankAccount** class contains the following constructor and member functions:

- void DisplayBalances();
  - Displays the account owners name and checking and savings balances.

```
Account Name....:     Mike
 Checking.......:   573.91
 Saving.........:  1025.00
```

- friend ostream& operator << (ostream&, BankAccount&);
  - A friend function that overloads the extraction (<<) operator for the ostream
  - This friend function will allow the account information for the ADT BankAccount to be printed directly (rather than using the DisplayBalances() function.
  - The information displayed by the friend functions should be identical to the DisplayBalances() function.

Bank Accounts – main() Code

```
/*************************************************************/
/*** Title: main.cpp
/*** Description: ban
/********************
#include <iostream>
#include <iomanip>
#include "BankAccount
using namespace std;

int main() {

    // Create Mike's
    BankAccount Mikes
    BankAccount Linda

    //***************
    // Mike's Account
    //***************

    // Display Mike's
    Mikes_Accts.Displ

    // Pay Day #1
    // Deposit $450 t
    Mikes_Accts.Depos
    Mikes_Accts.Depos
    Mikes_Accts.DisplayBalances();
```

```
    // Pay Some Bills
    Mikes_Accts.WithdrawChecking(39.50
    Mikes_Accts.WithdrawChecking(78.42
    Mikes_Accts.WithdrawChecking(103.1
    Mikes_Accts.DisplayBalances();

    // Pay Day #2
    // Deposit $250 to checking and $5
    Mikes_Accts.DepositChecking(250);
    Mikes_Accts.DepositSavings(50);
    cout << Mikes_Accts;

    // Rent is due, transfer Savings-t
    Mikes_Accts.TransferToChecking(500
    Mikes_Accts.WithdrawChecking(1205.
    cout << Mikes_Accts;

    // Pay Day #3
    // Deposit $675 to checking & tran
    Mikes_Accts.DepositChecking(675);
    Mikes_Accts.TransferToSavings(375)
    cout << Mikes_Accts;
```

```
    //***************************************//
    // Linda's Accounts
    //***************************************//

    // Display Linda's Accounts

    // Pay Day #1 for Linda
    //    -- Deposit $1450 to checking
    //    -- Deposit $800 to savings
    //    -- Display Linda's Accounts (using DisplayBalances() function)

    // Pay Some Bills for Linda
    //    -- Withdraw $148.57 from checking
    //    -- Withdraw $208.79 from checking
    //    -- Withdraw $1105.25 from checking
    //    -- Display Linda's Accounts (using the friend function)

    // Transfer Funds
    //    -- Transfer $850 from checking to savings
    //    -- Withdraw $575 from savings
    //    -- Display Linda's Accounts (using the friend function)

    return 0;
}
/*************************************************************/
```

Note: Download main.cpp from myCourses

Bank Accounts – Output

Microsoft Visual Studio Debug Console

```
Account Name....:     Mike
 Checking.......:   500.00
 Saving.........:  1000.00

Account Name....:     Mike
 Checking.......:   950.00
 Saving.........:  1100.00

Account Name....:     Mike
 Checking.......:   728.91
 Saving.........:  1100.00

Account Name....:     Mike
 Checking.......:   978.91
 Saving.........:  1150.00

Account Name....:     Mike
 Checking.......:   273.91
 Saving.........:   650.00

Account Name....:     Mike
 Checking.......:   573.91
 Saving.........:  1025.00
```