

PROBLEM STATEMENT:

The creation of a simple hierarchy of classes along with implementation and use of inheritance and use of an array of objects.

Notes:

This exercise builds on the Java program that you completed for Lab 5. You must use your final program from Lab Exercise 5 as the starting point for this exercise. You will also want to use code from Lab Exercise 3B.

This exercise correlates with **Chapter X** of the textbook.

The class hierarchy of this lab exercise program looks like the following:

LEVEL 1: LOAN

LEVEL 2: RESIDENTIAL, COMMERCIAL, AGRICULTURAL, UNSPECIFIED

You should be able to explain where inheritance is used in this Java program.

You should be able to explain where method overriding takes place in this program.

You should be able to explain the difference between the visibility modifiers.

CODE :

Add copies of the files from the project that you completed for Lab Exercise 5

Modify the Main driver class from the previous lab exercise as described below.

Within the while loop, add code to prompt the user for the type of Loan and input the type from the user. Be sure to present the list of available types to the user, namely (1) Residential Loan, (2) Commercial Loan, (3) Agricultural Loan, and (4) Unspecified Loan. Tell the user what to enter to specify the loan type. For example, you could show the numbers for the list loans, and tell the user to enter the number of their choice. An alternative would be to have the user enter the first letter of the type name.

The existing driver program includes the following statement to create the loan object (instance).

```
ln = new Loan(borrowerNameStr, loanId, propertyCost, downPayment, numYears);
```

Eliminate this statement. Replace the statement with code that creates the correct type of loan object (instance) based on the loan Residential type that the user entered. The program should create instances of the ResidentialLoan, CommercialLoan, AgriculturalLoan, and UnspecifiedLoan classes. The program should not create instances of the Loan class.

In Lab 6A you will only implement the ResidentialLoan class, and in Lab 6B you will implement the remaining three classes.

Modify the for-loop that creates the final display list of all the loans of all types. This for-loop is after the while(true) loop and iterates through only the actual (non-null) loan instances created. In order to accomplish this, do the following:

In the body of the for-loop, in the statement that concatenates to the variable outputStr2, add code to also display the loan type on a separate line. Use the following code to concatenate the loan type into your outputStr2 string:

```
... + "Loan Type: " + ln.getClass().getName() + ...
```

Be sure that your display includes all the data on each loan. For display purposes, combine some information on one line as shown below in the screen displays.

Add the code to display the total amount paid to the lender for the loan using the method computeTotalPaid that you will add to the loan class.

Keep the statement that appears after the for-loop and uses a JOptionPane dialog to display the contents of outputStr2.

Modify your Loan class. We will use the Loan class as the top class of our two-level class

hierarchy. Make the following modifications to your Loan class:

For each member variable, change the visibility (access) modifier from private to protected. This will enable the member variables to be inherited by subclasses of the Loan class.

Modify the toString() method to include the following code to add the name of the class to the returned string:

```
. . . + "Loan Type: " + this.getClass().getName() + . . .
```

Note: You will answer questions about the getClass method and the getName method in a later step of Exercise B.

Add a method to the loan class called computeTotalPaid(). It returns the total amount repaid to the lender for the loan (principle + interest).

Create a new public class, called ResidentialLoan, that is a subclass of Loan. This new class must be in a separate file with the same name as the new class. You can create the new class in the same package (directory/folder) that contains the Main class (file) and Loan class.

The ResidentialLoan class must inherit all the member variables of the Loan class. Do not declare any member variables in the ResidentialLoan class.

The ResidentialLoan class must have a public constructor that takes a value for each of the inherited member variables as parameter. The inherited variables are: name, loan ID, property cost, down payment number of years and annual interest rate(5.75%). The constructor must invoke the constructor of the parent class to initialize the variables. The ResidentialLoan constructor should not initialize the variables within its own body. The current residential rate is 5.75%.

The ResidentialLoan class must have a public computeMonthlyPayment method that calculates the monthly payment by simply invoking the computeMonthlyPayment method in the parent class after implementing the following specifications (1) for interest rates specified in a prior lab and (2) for a new specification: the residential loan must not exceed 30 years. If it does, reset the number of years to 30. That is, ResidentialLoans yield their typical cost but be limited to a 30 year maximum. Prior lab #3 interest rates: The current rate is 5.75%. However, if the down payment is less than 20% of the property cost, the residential interest rate is then 6.75%.

The ResidentialLoan class must have a public toString method that concatenates together and returns the following:

The string returned by invoking the parent class toString method to get a string that includes the values of all the member variables for a ResidentialLoan instance.

Note: The type of the loan should be included in the string returned by the parent class toString method. The result of formatting the return value from invoking the computeMonthlyPayment method that is within the ResidentialLoan class.

Note: In Part B of this exercise, you will create three additional new public classes, called CommercialLoan, AgriculturalLoan, and UnspecifiedLoan that are also subclasses of Loan.

Compile, execute, and debug the program. Test and debug your program using a variety of input data. Test using only the implemented ResidentialLoan class.

Run the program and make window captures of the dialogs displaying input / output.

Create batch file.

Create a batch file as you did for the previous assignments so that your program can be executed without the use of NetBeans.  
Be sure that the batch file is within your top level NetBeans project folder.

As usual create ReadMe.pdf file.

Create a file named ReadMe.pdf

In this document, insert your name at the top, and on the next line insert the assignment number

Then enter any comments regarding the assignment and your program.

Then insert several window captures of windows showing the inputs and outputs from the execution of the program.

Be sure the ReadMe file is within your top level project folder.

Zip the project folder and all its contents.

Change the name the zip file so that its name consists of your name along with the assignment number, as follows: " LastName\_ Lab\_06A\_cs209.zip".

Do not use spaces in the name of the file, use underscores or hyphens instead.

#### Deliverables:

Send to [duttat@ecc.edu](mailto:duttat@ecc.edu) an email this the exact subject

cs209\_ Lab\_06A

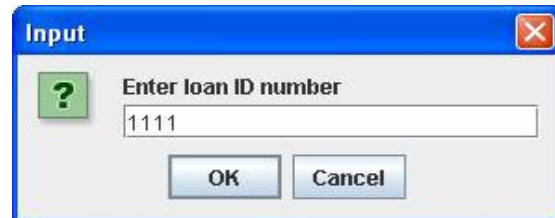
In this email attached the above named zip file

LastName\_ Lab\_06A\_cs209.zip

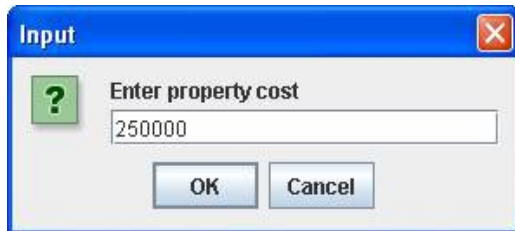
Due Date : 5:00pm 9 October 2014



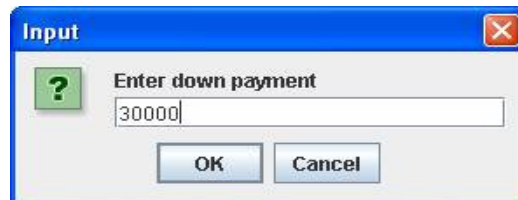
Input dialog box titled "Input" with a green question mark icon. The text "Enter Borrower name" is displayed above a text input field containing "John Smith". Below the input field are "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a green question mark icon. The text "Enter loan ID number" is displayed above a text input field containing "1111". Below the input field are "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a green question mark icon. The text "Enter property cost" is displayed above a text input field containing "250000". Below the input field are "OK" and "Cancel" buttons.



Input dialog box titled "Input" with a green question mark icon. The text "Enter down payment" is displayed above a text input field containing "30000". Below the input field are "OK" and "Cancel" buttons.

**Input**

Enter number of years for loan

35

OK Cancel

**Input**

Enter loan type by choosing R,C,A or U

R Residential Loan  
C Commercial Loan  
A Agricultural Loan  
U Unspecified Loan

Note: default loan is unspecified loan

R

OK Cancel

**Confirm**

1. Loan Type: lab06\_v3.ResidentialLoan, Name: John Smith ID: 1111, Cost: \$250000.00 Down Payment: \$30000.00, Years: 30, Rate: 6.75% , Monthly Payment \$1426.92

Max number of allowed Loans is: 7

Continue?

Yes No

**Results**

1. Loan Type: lab06\_v3.ResidentialLoan, Name: John Smith ID: 1111, Cost: \$250000.00 Down Payment: \$30000.00, Years: 30, Rate: 6.75% , Monthly Payment \$1426.92

2. Loan Type: lab06\_v3.ResidentialLoan, Name: David Smith ID: 2222, Cost: \$250000.00 Down Payment: \$30000.00, Years: 30, Rate: 6.75% , Monthly Payment \$1426.92

3. Loan Type: lab06\_v3.ResidentialLoan, Name: Tom Smith ID: 3333, Cost: \$250000.00 Down Payment: \$50000.00, Years: 30, Rate: 5.75% , Monthly Payment \$1167.15

4. Loan Type: lab06\_v3.ResidentialLoan, Name: Andrew Smith ID: 4444, Cost: \$250000.00 Down Payment: \$30000.00, Years: 25, Rate: 6.75% , Monthly Payment \$1520.01

5. Loan Type: lab06\_v3.ResidentialLoan, Name: Ellen Smith ID: 5555, Cost: \$250000.00 Down Payment: \$50000.00, Years: 25, Rate: 5.75% , Monthly Payment \$1258.21

6. Loan Type: lab06\_v3.ResidentialLoan, Name: Jane Smith ID: 6666, Cost: \$250000.00 Down Payment: \$20000.00, Years: 25, Rate: 6.75% , Monthly Payment \$1589.10

7. Loan Type: lab06\_v3.ResidentialLoan, Name: Tracy Smith ID: 7777, Cost: \$250000.00 Down Payment: \$60000.00, Years: 25, Rate: 5.75% , Monthly Payment \$1195.30

Program cannot handle any more additional Loans.

OK

Loan Summary





Loan list:

1. John Smith ID No: 1111

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$220000.00

Number Of Years: 30 Annual Interest Rate: 6.75%

Monthly Loan Payment: \$1426.92 Total Amount Paid for loan: \$513689.69

2. David Smith ID No: 2222

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$220000.00

Number Of Years: 30 Annual Interest Rate: 6.75%

Monthly Loan Payment: \$1426.92 Total Amount Paid for loan: \$513689.69

3. Tom Smith ID No: 3333

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$200000.00

Number Of Years: 30 Annual Interest Rate: 5.75%

Monthly Loan Payment: \$1167.15 Total Amount Paid for loan: \$420172.46

4. Andrew Smith ID No: 4444

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$220000.00

Number Of Years: 25 Annual Interest Rate: 6.75%

Monthly Loan Payment: \$1520.01 Total Amount Paid for loan: \$456001.61

5. Ellen Smith ID No: 5555

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$200000.00

Number Of Years: 25 Annual Interest Rate: 5.75%

Monthly Loan Payment: \$1258.21 Total Amount Paid for loan: \$377463.84

6. Jane Smith ID No: 6666

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$230000.00

Number Of Years: 25 Annual Interest Rate: 6.75%

Monthly Loan Payment: \$1589.10 Total Amount Paid for loan: \$476728.95

7. Tracy Smith ID No: 7777

Loan Type: lab06\_v3.ResidentialLoan

Property Cost: \$250000.00 Loan Amount: \$190000.00

Number Of Years: 25 Annual Interest Rate: 5.75%

Monthly Loan Payment: \$1195.30 Total Amount Paid for loan: \$358590.65

OK