ECE25100 Object Oriented Programming
Lab 2: Control Structures

Description:

The purpose of this lab is to help you understand how to

- perform simple computations,
- format numbers for display purposes,
- and make use of the **control** structures.

To get credit for the lab, you need to demonstrate to the instructor/lab assistant that you have completed all the requirements, and sign your name on the grade sheet.

QUESTION 1:

We will use the Java programming language to write a command line calculator. The program will ask the user which operation he wants to perform, then it will prompt the user for a first operand, and what is the second operand. It will then display the answer. The program will continue asking the user to use the calculator until the user chooses the "Exit". Note that in the case of a division, the program should make sure that the user is not attempting to do a division by zero. If that is the case, it should display an error message. Three examples of expected runs of the program are shown below:

```
Example 1
=========================================================
Welcome to Purdue University Northwest's calculator!

1 - Addition
2 - Subtraction
3 - Multiplication
4 - Division
5 - Exit

Which operation do you want to perform: 1
What is your first operand: 5
What is your second operand: 7

The result is 12.

Thanks for using Purdue University Northwest's calculator!
=========================================================
Example 2
=========================================================
Welcome to Purdue University Northwest's calculator!

1 - Addition
2 - Subtraction
3 - Multiplication
4 - Division
5 - Exit

Which operation do you want to perform: 4
What is your first operand: 3
```

```
What is your second operand: 0

Division by zero is not allowed.

Thanks for using Purdue University Northwest's calculator!
=========================================================
Example 3
=========================================================
Welcome to Purdue University Northwest's calculator!

1 - Addition
2 - Subtraction
3 - Multiplication
4 - Division
5 - Exit

Which operation do you want to perform: 5

Thanks for using Purdue University Northwest's calculator!
=========================================================
```

QUESTION 2:

PART A: `LoanCalculator` Requirements

Step 1: Create a **`LoanCalculator`** class with the following structure.

```
/*This class provides the service of calculating the payment necessary to
pay off a loan */
public class LoanCalculator {

public double monthlyPayment( double principle,
                              double interestRate, int term) {
   //missing code....
}

public void displaySchedule( double principle,
                             double interestRate, int term) {
  //missing code...
}

}//end of class LoanCalculator
```

Step 2: The interestRate parameter should be the annual interest rate expressed as a double between 0.0 and 100.0 (e.g. 5.0 for 5%).

Step 3: Provide the `public double monthlyPayment(double principle, double interestRate, int term)` method that returns the dollar amount that would be required to pay off the principle at the given interest rate over the term. If A is the principle, r the monthly interest rate as a fraction (i.e. interestRate/100/12)and t the term in months then the monthly payment is given by the formula:

monthlyPayment $= A * r * (r + 1)^t / ((r + 1)^t - 1)$

Step 4: The principle should be the amount of money borrowed and the term should be the number of months over which the loan is to be paid off.

Step 5: Provide the `public void dislaySchedule(...)` method which will display on `System.out` the repayment schedule of the loan. See the sample output for the program shown below.

Step 6: The `displaySchedule()` method should show the total amount of money paid (see sample output below).

Step 7: All money values should be shown with two decimal places (e.g. $100.00) and a $ sign.

Step 8: All interest rates should be shown as a decimal number between 0.00 and 100.00, should have at most two decimal places showing and should end in a % sign.

Step 9: If the `displaySchedule()` method requires the monthly payment amount if should invoke the `monthlyPayment()` method. That is, it should not have duplicate code that duplicates the calculation.

PART B: LoanApplication Requirements

Step 1: Provide a class **LoanApplication** with the following structure

```
/*This class runs the application and interacts with the user. It prompt the
user for input and displays results back to them on the console window */
import java.util.*;

public class LoanApplication{
public static void main(String[] args){

/*define the variables that will be used and their type for reading from the
keyboard*/
Scanner scanner;
GregorianCalendar today;
double principle; //amount of the loan
int term; //length of loan in months
double annualInterestRate; //loan interest rate

//create a Date object representing today's date
today = new GregorianCalendar();
System.out.println("Today is: "+today.getTime());

//create scanner for reading user input from keyboard
scanner = new Scanner(System.in);

//create the LoanCalculator object
LoanCalculator calculator = new LoanCalculator();

//set scanner to read a whole line at a time
String lineSeparator = System.getProperty("line.separator");
scanner.useDelimiter(lineSeparator);

//print greeting...missing code
//collect user input ...missing code
//compute monthly payment ...missing code
//display repayment schedule...missing code
}
}
```

Step 2: When the main method is run it should produce an output similar to the following.

```
Welcome to the Purdue University Northwest Loan Calculator
Today is: Fri Jan 26 12:25:50 CST 2018
=====================================

Enter the amount of the loan (e.g. 5000.00): 2000
Enter the annual interest rate: (e.g. 7.0): 6
Enter the term in months: (e.g. 60): 12

monthly payment will be: $172.13

The loan is $2000.00 at an interest rate of 6.00%

Month Payment Amount Remaining
1     $172.13 $1837.87
2     $172.13 $1674.92
3     $172.13 $1511.17
4     $172.13 $1346.59
5     $172.13 $1181.19
6     $172.13 $1014.96
7     $172.13 $847.90
8     $172.13 $680.01
9     $172.13 $511.28
10    $172.13 $341.70
11    $172.13 $171.28
12    $172.13 $0.00

The total amount paid was $2065.59
```

Step 3: The greeting should contains the current date.

Step 4: The user should be prompted for the principle, interest rate and desired term.

Step 5: The prompts should clearly tell the user what is being asked for and what format is expected. Use and example in the prompt itself as shown in the output above.

Step 6: There should be blank lines in the output where appropriate to make it more readable.

Step 7: All interaction with the user should be from the LoanApplication class. The LoanCalculator class methods should not be aware of the user or try to interact with them.

Step 8: The results shown by the program should be mathematically correct.

Step 9: Your program output test file should show several examples of your program working with different reasonable values of inputs. (We will not worry about unreasonable values yet in this assignment.)