CMM024 Object Oriented Programming

Practical Session: 2

1. Write a code to compare integers using if statements, relational and equality operators

Use six **if** statements to compare two integers input by the user. If the condition in any of these if statements is true, the statement associated with that if statement executes and display the fact.

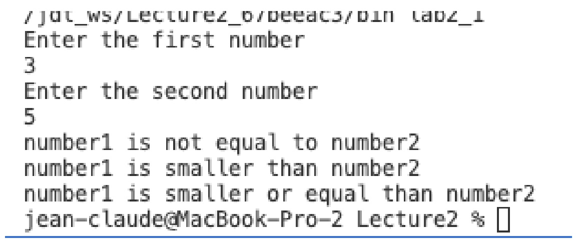
Note:

Each print or **println** statement resumes displaying characters from where the last print or **println** statement stopped displaying characters. Unlike **println**, after displaying its argument which adds an end of line, **print** does not position the output cursor at the beginning of the next line in the command window—the next character the program displays will appear immediately after the last character that print displays.

Tasks:

1) Create a **Lab2_1.java** java file, and create a **main** function
2) Create code to allow user to enter two integers using the **Scanner** and assign each of the values in two variables called number1 and number 2
3) Write 6 conditional statements to perform the following comparison operations (==, !=, >, <, >= and =<), and display a message when this is true.
4) Run the program

Output:



```
/jdt_ws/Lecture2_6/beeac3/bin Lab2_1
Enter the first number
3
Enter the second number
5
number1 is not equal to number2
number1 is smaller than number2
number1 is smaller or equal than number2
jean-claude@MacBook-Pro-2 Lecture2 % 
```

2. Write a code to analysis exam results using **while** loop and nested control statements

Note:

This is a simple program to demo how to use a while loop for data entry.

Explanations:

The control is simple. You enter 1 for Passes and 2 for Failures. Any other entry to exit the program. You accumulate the number of passes and failures in two variables as you go along inputting your scores. Then when the data entry is finished, you display the result.

Tasks:

1) Create a file java called **Lab2_2.java** and create the main function
2) Declare and create a Scanner object for user input
3) Declare two integer variables to hold values for passes and failures
4) Declare a boolean variable to control the while loop and set it appropriately.
5) In the while loop:
    a. first display "Enter result (1 - Pass, 2 = fail, Other value to Exit)", then get user input.
    b. Increase passes if 1 was entered, or failures if 2 was entered.
    c. If other value entered, display message that data entry is done and terminate the while loop.
6) Display results using the **printf** method for both variables when data entry is done.

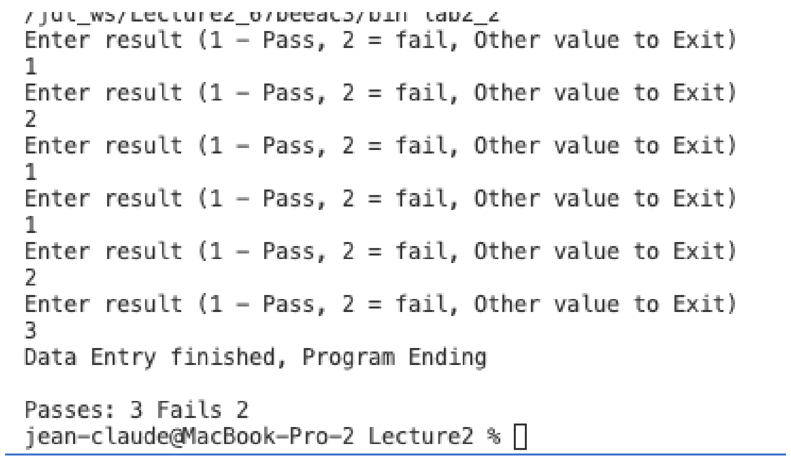
Hints / pseudo code

Use a boolean to check if you continue the entries or not. If you type 1 or 2 you do, otherwise that **boolean** should be set to false to terminate the loop. To check if you type a 1, 2 or 3, use an **if** statement. Check output below for more information.

```
Declare variables
```

```
Create user input (Scanner)
Declare and assign a value to boolean (continueInput )to control loop
While(continueInput)
    user types a number
    If number enter is 1
        Increase passes
    Else if number is 2
        Increase failures
    Else
        Display a message
        Set boolean continueInput to terminate while loop
Display number of passes and fails using printf
```

<u>Output:</u>



```
/jut_ws/Lecture2_6/beeac3/bin Lab2_2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
3
Data Entry finished, Program Ending

Passes: 3 Fails 2
jean-claude@MacBook-Pro-2 Lecture2 % 
```

3. Write a code to analysis exam results using **while** loop and nested control statements (2)

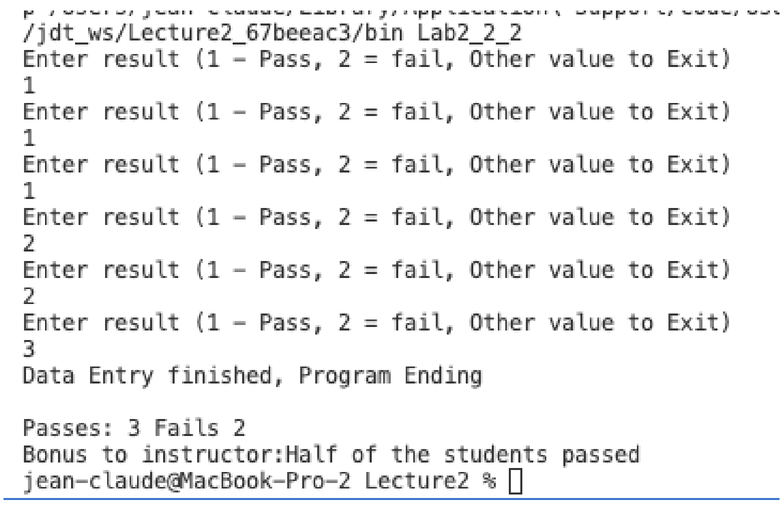
Use the code for question 2. The purpose of this exercise is to add extra code so you can use the percent of passes given several entries to display a message if the number of passes is over 50% of the number of entries.

<u>Hints / Pseudo Code:</u>

This is what needs to be added after the while loop etc.

```
double halfEntries = number of entries / 2;
if(passes > halfEntries)
    Display "Bonus to instructor: Half of the students passed"
```

Output:



```
[illegible]
/jdt_ws/Lecture2_67beeac3/bin Lab2_2_2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
1
Enter result (1 - Pass, 2 = fail, Other value to Exit)
2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
2
Enter result (1 - Pass, 2 = fail, Other value to Exit)
3
Data Entry finished, Program Ending

Passes: 3 Fails 2
Bonus to instructor:Half of the students passed
jean-claude@MacBook-Pro-2 Lecture2 % 
```

4. Write a code to calculate the Compound-Interest with **for** statement

Brief:

A person invests $5,00 in a savings account yielding 5% interest. Assuming that all the interest is left on deposit and no money is taken out of the account, calculate, and print the amount of money in the account at the end of each year for 5 years.

Use the following formula to determine the amounts:

$$a = p(1 + r)^n$$

where:

p is the original amount invested (i.e., often called the principal)

r is the annual interest rate (e.g., use 0.05 for 5%)

n is the number of years

a is the amount on deposit at the end of the nth year.

The solution to this problem should involve a loop that performs the indicated calculation for each of the 5 years the money remains on deposit. In method main declare double variables amount, principal and rate, and initialize principal to 500.0 and rate to 0.05. Java treats floating-point constants like 500.0 and 0.05 as type double.

Note:

Each year the principle increased.

Use the Math.pow (number1 , number2). This method is used to calculate a number (number1) raise to the power of some other number (number2) . This function accepts two parameters and returns the value of first parameter raised to the second parameter. Hence:

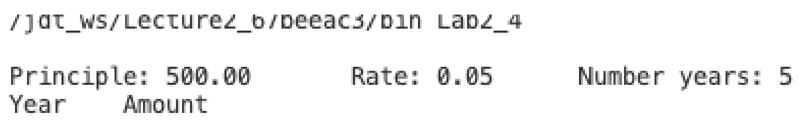
```
Math.pow(1.0 + r, n) will be code as:
Math.pow(1.0 + rate, year)
```

Start-up values:

- Principle = 500.0
- Rate = 0.05 (i.e., 5%)
- years = 5;

Tasks:

1) Create a java file called **Lab2_4.java** and create the main function
2) Declare 2 double variables to hold values for the principle and the interest rate, making sure that the variable holding that rate is a constant
3) Declare an integer variable to hold the number of years. Assign proper values to it.
4) Display using the printf method the values for the three variables. Use endline (\n) and tabs(\t) to format the output as shown below. Add another line for the headers (Here year and Amount), again using tabs etc.



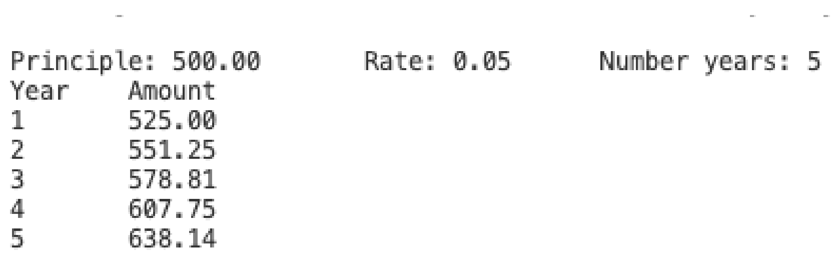
```
/jdt_ws/Lecture2_b/beeac3/bin Lab2_4

Principle: 500.00       Rate: 0.05      Number years: 5
Year    Amount
```

5) In the for loop, follow the pseudo code below. Note that we start with year = 1 and not 0 as this is for the following year!

```
For ( year = 1; year <= years; increment year)
      compute the new amount following given the formula above
      display the new amount.
```

Output:

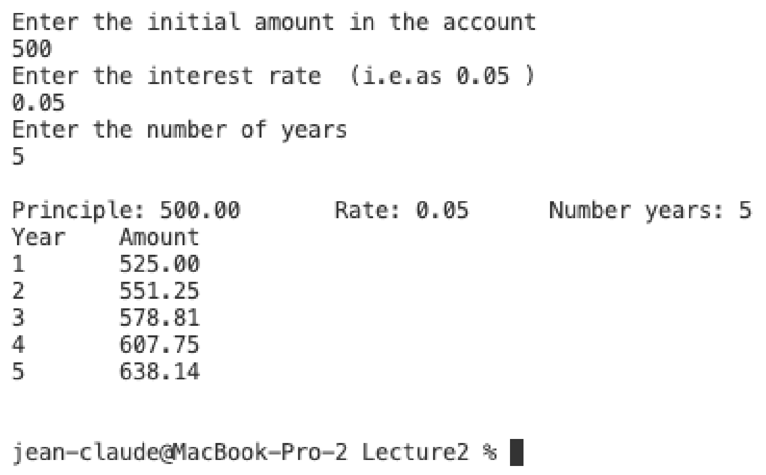


```
Principle: 500.00       Rate: 0.05      Number years: 5
Year    Amount
1       525.00
2       551.25
3       578.81
4       607.75
5       638.14
```

5. Write a code to calculate the Compound-Interest with **for** statement (2)

Use the code for question 4. Enhance your program so that the user can enter the principle, the rate, and the number of years before computing the results. Use the Scanner for this. Make sure you use nextDouble() and nextInt() when required.

<u>Output:</u>



```
Enter the initial amount in the account
500
Enter the interest rate  (i.e.as 0.05 )
0.05
Enter the number of years
5

Principle: 500.00       Rate: 0.05      Number years: 5
Year    Amount
1       525.00
2       551.25
3       578.81
4       607.75
5       638.14


jean-claude@MacBook-Pro-2 Lecture2 % █
```