GIFT School of Engineering and Applied Sciences

FAll 2020

CS-124: Introduction to Programming - Lab

Lab-4 Manual

Java Fundamentals – Part IV

v1.1
18/11/2020

Task #1: Writing mathematical expressions from Java code

In this task, you are being asked to understand the conversion of Java code into mathematical expressions.

Write the following Java expressions in mathematical notation on paper.

1. `dm = m * (Math.sqrt(1 + v / c) / Math.sqrt(1 - v / c) - 1);`
2. `volume = Math.PI * r * r * h;`
3. `volume = 4 * Math.PI * Math.pow(r, 3) / 3;`
4. `z = Math.sqrt(x * x + y * y);`

Task #2: Conversion into correct Java expressions

In this task, you are being asked to understand the conversion of mathematical expressions into Java code.

Write the following mathematical expressions in Java.

$$s = s_0 + v_0 t + \frac{1}{2} g t^2$$

$$\mathrm{FV} = \mathrm{PV} \cdot \left(1 + \frac{\mathrm{INT}}{100}\right)^{\mathrm{YRS}}$$

$$G = 4\pi^2 \frac{a^3}{p^2(m_1 + m_2)}$$

$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

1. Create a program called **JavaExpressionsLab4.java.**
2. Create appropriate variables and assign any values.
3. Convert all above formulae into correct Java expressions**.**
4. Note that **cos** function can be used as **Math.cos(one input value)**.
5. Print all the results of expression by giving appropriate messages.

Task #3: Writing and evaluating expressions

In this task, you are being asked to understand the evaluation order of various operators.

What are the values of the following expressions, assuming that **n** is **17** and **m** is **18**? First solve all expressions on paper, and then write and solve in Java.

Make sure that you truly understand why and how each expression gets solved and produces the result.

a. `n / 10 + n % 10`

b. `n % 2 + m % 2`

c. `(m + n) / 2`

d. `(m + n) / 2.0`

e. `(int) (0.5 * (m + n))`

f. `(int) Math.round(0.5 * (m + n))`

Now, crate a program called **ExpressionsLab4.java,** and write output statements and print both the expression and its value.

Task #4: Identifying and correcting compile-time errors

In this task, you are being asked to find and correct compile-time errors in a code.

Find at least **five** *compile-time* errors in the following program.

```
public class HasErrors {
  public static void main();
  {
    System.out.print(Please enter two numbers:)
    x = in.readDouble;
    y =  in.readDouble;
    System.out.printline("The sum is " + x + y);
  }//main
}//class
```

Now, create a program that has the correct code. Execute and display the output using appropriate messages.

Task #5: Using Math class functions

In this task, you are being asked to apply Math class functions.

Write a program that computes the first and last digit of a number. For example, if the input is **23456**, the program should print **2** and **6**.

Hint: Use **%** and **Math.log10**.

1. Create a program called **MathClassLab4.java.**
2. Create an integer variable and take input from the user using a **Scanner** object**.**
3. Finally, print the input number, and it's first and last digits using appropriate messages.

Task #6: Convert word problem into Java code

In this task, you are being asked to write a given word problem as a program.

Write a program that displays the dimensions of a letter-size (8.5 × 11 inches) sheet of paper in millimeters. There are 25.4 millimeters per inch. This program also computes and displays the perimeter of this letter-size (8.5 × 11 inches) sheet of paper and the length of its diagonal.

HINT: Perimeter of a rectangle = **2 x (length + width)**, and diagonal = **Square Root of (length^2 + width^2).**

NOTE: Use constants and comments in your program.

1. Create a program called **InchesToMillimetersLab4.java.**
2. Create appropriate variables with values as given in the statement above. You should create a constant for the total millimeters per inch.
3. Finally, print the required values using appropriate messages.

Task #7: Convert word problem into Java code

In this task, you are being asked to write a given word problem as a program.

Write a program that prompts the user for a radius and then prints:

- The area and circumference of a circle with that radius
- The volume and surface area of a sphere with that radius

HINT: Area = $\pi \times radius^2$, Circumference$=2 \times \pi \times radius$,

Volume of sphere = $(4/3) \times \pi \times radius^3$, Surface area = $4 \times \pi \times radius^2$.

NOTE: Use Math class constants and functions in your program.

1. Create a program called **CircleSphereLab4.java.**
2. Create appropriate variables with values as given in the statement above, using a **Scanner** object.
3. Finally, print the required values using appropriate messages.

Task #8: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

Write a program that asks the user to input (using a **Scanner** object):

- The number of gallons of fuel in the tank
- The fuel efficiency in miles per gallon
- The price of fuel per gallon

Then print the cost per 100 miles and how far the car can go with the gas in the tank.

1. Create a program called **FuelEfficiencyLab4.java.**
2. Create appropriate variables with values as given in the statement above, using a **Scanner** object.
3. Finally, print the required values using appropriate messages.

Task #9: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

Write a program that reads in an integer and breaks it into a sequence of individual digits. For example, the input **16384** is displayed as:

1 6 3 8 4

You may assume that the input has no more than five digits and is not negative.

1. Create a program called **DigitsLab4.java.**
2. Create an appropriate variable with values as given in the statement above, using a **Scanner** object.
3. Run and print the required value.

Task #10: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

Write a program that reads two times in military format (0900, 1730) and prints the number of hours and minutes between the two times. Here is a sample run. User input is in color.

```
Please enter the first time: 0900
Please enter the second time: 1730
8 hours 30 minutes
```

Extra credit if you can deal with the case where the first time is later than the second:

```
Please enter the first time: 1730
Please enter the second time: 0900
15 hours 30 minutes
```

1. Create a program called **TimeDiffLab4.java.**
2. Create appropriate variables with values as given in the statement above, using a **Scanner** object.
3. Run and print the required values.

Task #11: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

Easter Sunday is the first Sunday after the first full moon of spring. To compute the date, you can use this algorithm, invented by the mathematician Carl Friedrich Gauss in 1800:

1. Let y be the year (such as 1800 or 2001).
2. Divide y by 19 and call the remainder a. Ignore the quotient.
3. Divide y by 100 to get a quotient b and a remainder c.
4. Divide b by 4 to get a quotient d and a remainder e.
5. Divide 8 * b + 13 by 25 to get a quotient g. Ignore the remainder.
6. Divide 19 * a + b - d - g + 15 by 30 to get a remainder h. Ignore the quotient.
7. Divide c by 4 to get a quotient j and a remainder k.
8. Divide a + 11 * h by 319 to get a quotient m. Ignore the remainder.
9. Divide 2 * e + 2 * j - k - h + m + 32 by 7 to get a remainder r. Ignore the quotient.
10. Divide h - m + r + 90 by 25 to get a quotient n. Ignore the remainder.
11. Divide h - m + r + n + 19 by 32 to get a remainder p. Ignore the quotient.

Then Easter falls on **day p of month n**. For example, if **y** is 2001:

```
a = 6     g = 6     m = 0     n = 4
b = 20,   c = 1     h = 18    r = 6     p = 15
d = 5,    e = 0     j = 0,    k = 1
```

Therefore, in 2001, Easter Sunday fell on April 15. Write a program that prompts the user for a year and prints out the month and day of Easter Sunday.

1. Create a program called **EasterSundayLab4.java.**
2. Create an appropriate variable for the year and ask the user for an input using a **Scanner** object.
3. Run and print the required values using appropriate messages.

Task #12: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

The dew point temperature T_d can be calculated (approximately) from the relative humidity RH and the actual temperature T by

$$T_d = \frac{b \cdot f(T, RH)}{a - f(T, RH)}$$

$$f(T, RH) = \frac{a \cdot T}{b + T} + \ln(RH)$$

where $a = 17.27$ and $b = 237.7°$ C.

Write a program that reads the relative humidity (between 0 and 1) and the temperature (in degrees C) and prints the dew point value. Use the Java function `log` to compute the natural logarithm.

1. Create a program called **DewPointTempLab4.java.**
2. Create appropriate variables with values as given in the statement above, using a **Scanner** object.
3. Use **Math.log(one input value)** for finding the natural logarithm.
4. Run and print the required values.

Task #13: Converting word problems into programs

In this task, you are being asked to write a given word problem as a program.

The pipe clip temperature sensors shown here are robust sensors that can be clipped directly onto copper pipes to measure the temperature of the liquids in the pipes.

Each sensor contains a device called a *thermistor*. Thermistors are semiconductor devices that exhibit a temperature-dependent resistance described by:

$$R = R_0 \, e^{\beta\left(\frac{1}{T}-\frac{1}{T_0}\right)}$$

where R is the resistance (in Ω) at the temperature T (in °K), and R_0 is the resistance (in Ω) at the temperature T_0 (in °K). β is a constant that depends on the material used to make the thermistor. Thermistors are specified by providing values for R_0, T_0, and β.

The thermistors used to make the pipe clip temperature sensors have $R_0 = 1075\ \Omega$ at $T_0 = 85$ °C, and $\beta = 3969$ °K. (Notice that β has units of °K. Recall that the temperature in °K is obtained by adding 273 to the temperature in °C.) The liquid temperature, in °C, is determined from the resistance R, in Ω, using

$$T = \frac{\beta T_0}{T_0 \ln\left(\frac{R}{R_0}\right) + \beta} - 273$$

Write a Java program that prompts the user for the thermistor resistance R and prints a message giving the liquid temperature in °C.

1. Create a program called **LiquidTempLab4.java.**
2. Create appropriate variables with values as given in the statement above, using a **Scanner** object.
3. Use **Math.log(one input value)** for finding the natural logarithm.
4. Run and print the required values.