EECS 1022 Labtest 1 Version A

Friday October 19, 2018, 10:00-11:30

Instructions

This labtest is closed book, no aids allowed. You must work individually. You are not allowed to browse any documents on the web other than those provided in this labtest page.

There is a set of multiple-choice questions and a set of programming questions. You will fill your answers to all these questions into a Java class called `Utilities.java` which you will then submit. Here is how to proceed:

- First download the starter code for the Utilities.java class and save it on the Desktop.
- Then, also download the code for the UtilitiesTester.java class and save it on the Desktop.
- Then launch Android Studio and start a new project called `Labtest1A` (choose "no activity" and accept other options).
- Then in this project, create a new Java class `Utilities.java` within package `eecs1022.labtest1a`. After the class opens in the editor window, paste in the starter code in the `Utilities.java` file that you downloaded earlier and save.
- After this, create another new Java class `UtilitiesTester.java` within package `eecs1022.labtest1a` in the project, and also paste in the code in the `UtilitiesTester.java` file that you downloaded earlier and save.
- Then go over the questions below and enter your answers into the `Utilities.java` file and save your changes.
- Test your solutions by running the `UtilitiesTester.main()` method (right click on the `UtilitiesTester.java` class in the project window and select to run `UtilitiesTester.main()`). Your output should be as shown here.
- When you are ready, use Web Submit to submit your modified `Utilities.java`. To do this, first open the link https://webapp.eecs.yorku.ca/submit in your browser, login using your Passport York account, select your course `1022`, select the assignment `labtest1a`, navigate to and choose the file `Utilities.java` (it should be in folder `AndroidStudioProjects/Labtest1A/app/src/`), and then click the "Submit files" button at the bottom of the page. If you have successfully completed this task, you will see a message saying that you have submitted the file `Utilities.java` with the submission time.
- You can submit as many times as wou want and only the last version submitted will be graded.
- **You must submit your answers before the labtest period ends at 11:30; submission will be disabled after that time.**
- **Make sure that your `Utilities.java` class compiles without errors. If there are compilation errors, you may get a grade of 0.**

If you finish early, you may leave the lab. After the labtest ends, the lab will remain open until the end of the lab period and you may start working on Lab 3, which is described in Chapter 3 A Symphony of APIs - Doing of the textbook.

Multiple Choice Questions

For these questions, enter your answers into the `Utilities.java` file by editing the method corresponding to the question, replacing the `?` in the printed string by your answer, a, b, c, d, or e.

Question 1

Suppose that we have an `int` value *val* and that we want to assign it to a variable `var`. The assignment cannot be performed without doing a type cast if the type of `var` is?

a. `long`,
b. `float`,
c. `double`,
d. `short`,
e. none of the above.

Question 2

In the following Java arithmentic expression
`a + b % c - d / e * f`
which operation is performed first?

a. `+`,
b. `%`,
c. `-`,
d. `/`,
e. `*`.

Question 3

Suppose that we have an `int` variable `ivar` that has been initialized to `0`. Which of the following assignments will lead to an `ArithmeticException` being thrown?

a. `boolean b = 3 / ivar < 100 && ivar !=0;`
b. `boolean b = ivar !=0 && 3 / ivar < 100;`
c. both of the above,
d. `boolean b = 3 / (ivar + 1) < 100;`
e. none of the above.

Question 4

Which Java primitive whole number type does not admit negative values?

a. `long`,
b. `int`,
c. `short`,
d. `char`,
e. `byte`.

Question 5

Suppose that we have an `int` variable `k` that has been initialized. Which of the following assignments does not increment `k` by 1?

a. `k++;`
b. `k = k + 1;`
c. `k += 1;,`
d. `k = k + 1/2 + 1/2;,`
e. `++k;`

Programming Questions

Question 6

Go to the `Utilities.java` class and implement the method

```
double surfaceOfSphere(double radius)
```

Given the radius r of a sphere, return its surface, using the formula $4 \pi r^2$, where you must use the value $\pi = 3.14$.

You can assume that all input values of `radius` are greater than or equal to 0. There is no need to check for it.

Your returned double value may be within ± 0.1 of the expected output.

You can see some examples in the `UtilitiesTester.java` class and expected output `UtilitiesTesterOutput.txt`.

Question 7

In the `Utilities.java` class, implement the method

```
double farenheit2Celsius(double degrees)
```

Given a temperature in t degrees Farenheight, return the equivalent temperature in degrees Celsius, using the formula $5/9\ (t - 32)$.

It is not required to validate the input `degrees`.

Your returned double value may be within ± 0.1 of the expected output.

You can see some examples in the `UtilitiesTester.java` class and expected output `UtilitiesTesterOutput.txt`.

Question 8

In the `Utilities.java` class, implement the method

```
double getBMI(int pounds, int feet, int inches)
```

Given a weight in pounds and a height in feet and inches, it returns the person's body mass index (BMI).

You must convert the weight in pounds into kilograms by using the following conversion rate: 1 pound is equal to 0.453592 kilograms.

You must convert the height in feet and inches into metres by using the following conversion rates: 1 foot is equal to 0.3048 metre, and 1 inch is equal to 0.0254 metre.

Given a weight in kilograms w and a height in metres h, the BMI is w / h^2.

You can just return the BMI value as a double; you don't need to format it.

You can assume that the input arguments are positive integers; it is not necessary to validate them.

Your returned double value may be within ± 0.1 of the expected output.

You can see some examples in the `UtilitiesTester.java` class and expected output `UtilitiesTesterOutput.txt`.

Question 9

In the `Utilities.java` class, implement the method

```
boolean isSortedSequence(int n1, int n2, int n3, int n4, int n5)
```

Given five integer values, do they form an ascending sequence?

It is a (non-decreasing) sorted sequence if for every two consecutive numbers, the left number is less than or equal to the right number, e.g., (1, 3, 3, 4, 5).

You can see more examples in the `UtilitiesTester.java` class and expected output `UtilitiesTesterOutput.txt`.

Question 10

In the `Utilities.java` class, implement the method

```
boolean isLeapYear(int year)
```

Given an argument `year`, the method returns true if and only if it is a leap year, i.e., has 366 days instead of the usual 365.

A year y is a leap year if y is a multiple of 4 and is not a multiple of 100, unless it is a multiple of 400.

You can assume that `year` is a positive integer. There is no need to check for it.

You can see some examples in the `UtilitiesTester.java` class and expected output `UtilitiesTesterOutput.txt`.
