

EF234302 Object-Oriented Programming (E)**Quiz 1**

Starting date:	19 September 2024
Deadline:	26 September 2024, 23:59 WIB. Penalty: 0.15% of grade/minute of tardiness.
Exam type:	Open, Individual Project
Send to:	MM Irfan Subakti <yifana@gmail.com> CC to Muhammad Danis Hadriansyah <mdanishadriansyah@gmail.com> with the subject: EF234302_OOP(E)_Q1_StudentID_Name
File type and format:	A zip file containing all of the .java source files & the declaration
Filename format:	EF234302_OOP(E)_Q1_StudentID_Name.ZIP

Instruction

Please do these steps as in the following.

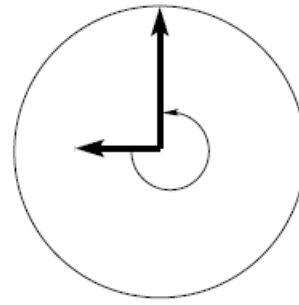
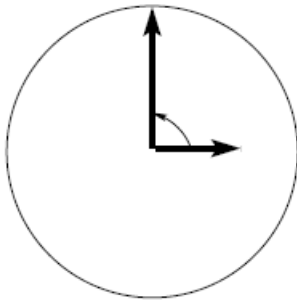
1. **[10 points]** In the following e_1 and d_1 represent the fraction $\frac{e_1}{d_1}$ and correspondingly e_2 and d_2 the fraction $\frac{e_2}{d_2}$, where e_1 and e_2 are integers and d_1 and d_2 are positive integers. Write a Java class, namely `Q11Fraction.java`, which computes numbers es , ds , and ep , dp , which stand for the sum and the product of the two fractions, respectively. E.g., $\frac{1}{2} + \frac{1}{3} = \frac{5}{6}$, and $\frac{1}{2} * \frac{1}{3} = \frac{1}{6}$. Note that addition and multiplication on fractions are defined by:

$$\frac{e_1}{d_1} + \frac{e_2}{d_2} = \frac{e_1*d_2 + e_2*d_1}{d_1*d_2} \quad \frac{e_1}{d_1} * \frac{e_2}{d_2} = \frac{e_1*e_2}{d_1*d_2}$$

Test your class for $\frac{1}{2} + \frac{1}{3}$, $\frac{1}{3} + \frac{3}{4}$, $\frac{1}{2} * \frac{2}{3}$, and $\frac{1}{4} * \frac{2}{3}$.

```
// File: Q11Fraction.java
public class Q11 {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 1 above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class in here
    }
}
```

2. **[10 points]** This time is 13:03. We represent the time by two variables `hours` and `minutes`, that is `hours = 13;` and `minutes = 03;`. Write a Java class, namely `Q12Time.java`, which computes the angle between the hour hand and the minute hand on a traditional analog clock. Angles should be measured counterclockwise from hour to minute hand. The results should be rounded and normalized so that it is between 0 and 359 (inclusively). For instance, the angles at 3:00 and 9:00 hours should be 90° and 270°, respectively.



Hint: 1 minute $\approx 6^\circ$, 1 hour $\approx 30^\circ$. Start at 12 o'clock.

Note that your program must be able to compute the output for arbitrary inputs, in which hours may take values from 0 through 24 and minutes from 0 to 60. Do not forget to write comments which explain why your program is correct.

Test your program for the following times: 9:00, 3:00, 18:00, 1:00, 2:30, and 4:41 (with results 270° , 90° , 180° , 30° , 255° , and 254° , respectively).

```
// File: Q12Time.java
public class Q12Time {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 2 above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class in here
    }
}
```

3. [10 points] Define a class `Q13ComputerAccount` and a constructor to create it. A computer account should be built from the three strings `realName`, `userName`, and `password`. It should implement methods `printRealName`, `printUserName`, `printPassword`, and `changePassword`. The first three methods take no arguments. The last method takes one argument of type `String`, the new password, and sets the password to the new password; its return value is `void`. Note that you have always to comment and test your program appropriately, not just for this assignment. We will not write this to the assignments in the future, but still, if you fail to do so marks will be subtracted.

```
// File: Q13ComputerAccount.java
public class Q13ComputerAccount {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 3 above.
     * Remember how to comment properly.
     */
    // Define your properties/fields here

    // Define your constructor here

    // Define your getters and setters in here
}
```

```
// File: Q13ComputerAccountTest.java
public class Q13ComputerAccountTest {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 3 above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class: Q13ComputerAccount in here
    }
}
```

4. **[10 points]** Define a class `Q14ComplexNumber` as well as a constructor and getter methods. A `Q14ComplexNumber` should represent a complex number with real and imaginary parts. Implement these methods to:

1. Add two complex numbers,
2. Subtract two complex numbers,
3. Multiply two complex numbers,
4. Convert the complex number to a String "a + bi". (where a is the real part and b is the imaginary part).

Operations:

1. Additions = $(a + bi) + (c + di) = (a + c) + (b + d)i$
2. Subtractions = $(a + bi) - (c + di) = (a - c) + (b - d)i$
3. Multiplications = $(a + bi) * (c + di) = (a * c - b * d) + (a * d + b * c)i$

```
// File: Q14ComplexNumber.java
public class Q14ComplexNumber {
    /**
     * Write down in this class your solution and
     * its explanation of the question above.
     * Remember how to comment properly.
     */
    // Define your properties/fields here
    // Define your constructor here
    // Define your method/functions in here
}

// File: Q14ComplexNumberTest.java
public class Q14ComplexNumberTest {
    /**
     * Write down in this class your solution and
     * its explanation of the question above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class: Q14ComplexNumber in here
    }
}
```

5. **[20 points]** In Lecture #3: Exercises, Ex2. Conversion, we wrote a program that converts masses given in the imperial system into the metric system. Write a program that can deal with weights given in pounds. Make use of the conversion:

1 pound = 0.45359237 kilograms

Define a `Q15Weight` class, and write a constructor `Q15Weight(double p)` to generate a weight given in pounds. Implement methods `getPounds` and `getKilograms` which return the weight in pounds and kilograms, respectively.

```
// File: Q15Weight.java
public class Q15Weight {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 5 above.
     * Remember how to comment properly.
     */
    // Define your properties/fields here

    // Define your constructor here

    // Define your methods/functions in here
}
```

```
// File: Q15WeightTest.java
public class Q15WeightTest {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 5 above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class: Q15Weight in here
    }
}
```

6. **[20 points]** In no. 1 we looked at the addition and multiplication of two fractions. In this assignment, fractions should be represented by a Java class `Q16Fraction`. You have to define this class, which in addition to the constructor and the accessor methods `getNumerator` and `getDenominator` have methods `toString` (to print a rational number) as well as `getSum` and `getProduct`, which generate the sum and the product of fractions and return the corresponding object of Class `Q16Fraction`. For instance, if we generate fractions `Q16Fraction f1 = new Q16Fraction(1, 2);` and `Q16Fraction f2 = new Q16Fraction(3, 7);` then

- `f1.toString()` should return the string `"1/2"`;
- `System.out.print(f2.getProduct(f1));` should print `3/14` (which corresponds to the product of `f2` and `f1`); and
- `System.out.print(f2.getSum(f1));` should print `13/14` (which corresponds to the sum of `f2` and `f1`).

```
// File: Q16Fraction.java
public class Q16Fraction {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 6 above.
     * Remember how to comment properly.
     */
    // Define your properties/fields here

    // Define your constructor here

    // Define your methods/functions in here
}
```

```
// File: Q16FractionTest.java
public class Q16FractionTest {
    /**
     * Write down in this class your solution and
     * its explanation of question no. 6 above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class: Q16Fraction in here
    }
}
```

7. [20 points] Define two opposing classes `Player` and `Enemy`. Both classes should have name, health, power, and defense. Define a constructor, getters, and setters to those attributes. Implement these methods in both classes:

- `attack()`; this method should take the opposing class as its parameter. The opposing class should call the method below with a parameter of this class' attack:
- `takeDamage()`; this method takes the parameter of the opposing class's attack, and then this class' health will be reduced by (opposing.attack - this.defense). If this class' health is lower than zero, outputs " {this.name} died!"

```
// File: Player.java
public class Player {
    /**
     * Write down in this class your solution and
     * its explanation of the question above.
     * Remember how to comment properly.
     */
    // Define your properties/fields here
    // Define your constructor here
    // Define your method/functions in here
}

// File: Enemy.java
public class Enemy {
    /**
     * Write down in this class your solution and
     * its explanation of the question above.
     * Remember how to comment properly.
     */
}
```

```

    */
    // Define your properties/fields here
    // Define your constructor here
    // Define your method/functions in here
}
// File: EnemyPlayerTest.java
public class EnemyPlayerTest {
    /**
     * Write down in this class your solution and
     * its explanation of the question above.
     * Remember how to comment properly.
     */
    public static void main(String[] args) {
        // Test your class: Player and Enemy in here
    }
}

```

8. To avoid plagiarism/cheating, every student needs to pledge and declare, then she/he must submit her/his **signed pledge and declaration** as in the following. Failing to do so will result in getting a 0 (zero) grade. Attach the **scanned/photo** of your *declaration* to your report.

“By the name of Allah (God) Almighty, herewith I pledge and truly declare that I have solved quiz 1 by myself, didn’t do any cheating by any means, didn’t do any plagiarism, and didn’t accept anybody’s help by any means. I am going to accept all of the consequences by any means if it has proven that I have done any cheating and/or plagiarism.”

[Place, e.g., Surabaya], [date, e.g., 26 September 2024]

<Signed>

[Full name, e.g., Palupi Rantani]

[StudentID, e.g., 05112340000xxx]

9. Have a wonderful day, guys! Good luck! 😊