OOP Examples

- **1**) Define a class called *Rational*, which represents rational numbers with their arithmetic operations
- Use integer variables to represent the private instance variables of the class: numerator and denominator
- Call them **num** and **den**
- Write a **default constructor**, which initializes the numerator with *0* and the denominator with *1*
- Another **constructor** with two integer parameters will keep the fraction in its reduced form (for instance, the constructor with *2* and *4* as parameters will be stored into the object as *1/2*) – for this, define a separate **static** method for computing the greatest common divisor (you will also need this method later) between the numerator and the denominator and simplfy the fraction *2/4* with the greatest common divisor, in order to obtain *1/2*

- The class *Rational* will contain public methods which will implement the following operations on rational numbers:
- a) *public Rational add(Rational a, Rational b)* – addition of two rational numbers – the result will be a rational number stored in the reduced form
- $\frac{2}{3}+\frac{4}{5}=\frac{2*5+4*3}{3*5}=\frac{10+12}{15}=\frac{22}{15}$
- b) *public Rational sub(Rational a, Rational b)* – subtraction of two rational numbers – the result will be a rational number stored in the reduced form
- $\frac{14}{17}-\frac{5}{9}=\frac{14*9-5*17}{17*9}=\frac{126-85}{153}=\frac{41}{153}$

- *public Rational mult(Rational a, Rational b)* – multiplication of two rational numbers – the result will be a rational number stored in the reduced form
- $\frac{2}{3} * \frac{7}{10} = \frac{2*7}{3*10} = \frac{14}{30} = \frac{7}{15}$
- *public Rational div(Rational a, Rational b)* – division of two rational numbers – the result will be a rational number stored in the reduced form
- $\frac{\frac{4}{5}}{\frac{7}{10}} = \frac{4}{5} * \frac{10}{7} = \frac{4*10}{5*7} = \frac{40}{35} = \frac{8}{7}$

- *public void intFormat()* – displaying the rational number in the numerator/denominator format
- *public void realFormat()* – displaying the rational number in the floating point format

- Define another class, called *TestRational*, for testing the four operations defined in the *Rational* class, using the previous examples or other examples at your choice
- Display the results of the four operations using the methods *intFormat()* and *realFormat()*

- **2**) Define a class *Complex*, which represents complex numbers with their arithmetic operations
- Use double variables to represent the private instance variables of the class: real part and imaginary part
- Call them **re** and **im**
- The complex numbers have the form *realPart + i * imaginaryPart*, where $i^2 = -1$
- Define two constructors, the default one that initializes at *0.0* the instance variables and the other that initializes them with the values of the parameters

- Write public methods for the following operations:
- a) *public Complex add(Complex a, Complex b)* – addition of two complex numbers – the result is a complex number
- $(2 + 3 * i) + (4 + 5 * i) = 6 + 8 * i$
- b) *public Complex sub(Complex a, Complex b)* – subtraction of two complex numbers – the result is a complex number
- $(2 + 3 * i) - (4 + 5 * i) = -2 - 2 * i$

- *c) public Complex mult(Complex a, Complex b)* – multiplication of two complex numbers – the result is a complex number
- $(2+3\ast i)\ast(4+5\ast i)=2\ast 4+2\ast 5\ast i+3\ast 4\ast i+3\ast 5\ast i^2=8+10\ast \mathrm{i}+12\ast \mathrm{i}-15=-7+22\ast i$
- d) *public Complex div(Complex a, Complex b)* – division of two complex numbers – the result is a complex number
- $\frac{2+3\ast i}{4+5\ast i}=\frac{(2+3\ast i)\ast(4-5\ast i)}{(4+5\ast i)\ast(4-5\ast i)}=\frac{2\ast 4-2\ast 5\ast i+3\ast 4\ast i-3\ast 5\ast i^2}{4\ast 4-4\ast 5\ast i+4\ast 5\ast i-5\ast 5\ast i^2}=\frac{8-10\ast \mathrm{i}+12\ast \mathrm{i}+15}{16+25}=\frac{23+2\ast \mathrm{i}}{41}=\left(\frac{23}{41}\right)+\left(\frac{2}{41}\right)\ast \mathrm{i}=0.56097+0.04878\ast i$
- e) *public void tupleFormat()* – displaying a complex number in the format *(a,b)* where *a* is the real part and *b* is the imaginary part of the complex number

- Define another class, called *TestComplex*, for testing the four operations defined in the *Complex* class, using the previous examples or other examples at your choice
- Display the results of the four operations using the method *tupleFormat()*