

Name and, if possible, ID#: 7c

AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
COMP120 Introduction to Object-Oriented Programming

FINAL EXAM

Date: Monday, May 18 2015
Starting time: 09:20
Duration: 1 hour 40 minutes
Attention: ANY TYPE OF COMMUNICATION IS PROHIBITED
Please write down your name at the top of all used pages

1/15

1/15
F

Problem 1

Consider below a *public interface Valuable* that includes the only method *public double value(double x)*:

```
public interface Valuable {  
    public double value(double x);  
}
```

1.1 Implement a *public class Function* that encapsulates a member variable of type *Valuable* and computes its max in the specified range from x_1 to x_2 by looking at:

$f(x_1), f(x_1+dx), f(x_1+2dx), \dots, f(x_1+k*dx)$, where $k = 1, 2, \dots$ and $x_1+k*dx < x_2$

```
public class Function {  
    private Valuable f;  
    private double dx;  
  
    public Function(Valuable newValuable, double newDX) {  
        //TO BE IMPLEMENTED for(int c=0; Function(Valuable + c*dx) < Function(x2);  
        f = new Valuable(newValuable);  
        dx = new DX;  
    }  
    public double max(double x1, double x2) {  
        //TO BE IMPLEMENTED for(int c=0; Function(x1 + c*dx) < Function(x2);  
        return c;  
    }  
}
```

1.2 Implement an expression

$$a * \sin(x) + b * \cos(x)$$

as a *public class Harmonic* that implements the interface *Valuable* and encapsulates double parameters *a* and *b*. The parameters are initialized by the two-argument constructor *public Harmonic(double newA, double newB)*;

1.3 In a separate *public static void main(String args[])* write a code that inputs two double values, creates an object of type *Harmonic* and, using the class *Function*, prints its maximal value in the range from $x_1 = -1.5$ to $x_1 = 1.5$:

```
public static void main(String args[]) {  
    Scanner input = new Scanner(System.in);  
    double a = input.nextDouble(), b = input.nextDouble();  
  
    //TO BE COMPLETED
```

```
}  
public class Harmonic(double newA, double newB) {  
    return newA * sin(x) + newB * cos(x);  
}
```

3.

1

Use the backside, if needed

Page 1 of 4

OOP FT. 180515. M111