Name and, if possible, ID#

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

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:

$f(x_1), f(x_1+dx), f(x_1+2dx), \ldots, f(x_1+k*dx)$, where $k = 1, 2, \ldots$ 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
      }

      public double max(double x1, double x2) {
         //TO BE IMPLEMENTED
      }
```

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

}
```

Use the backside, if needed

① f = new Valuable;
dX = new DX;

② double X = X1, max = f.value(X1);
for(k=1, k<(X2-X1)/dX, k++){
if (f.value(X) < f.value(X+dX)
max = f.value(X+dX);
X += dX;
}
return max;

1.2| public class Harmonic implements Valuable{
private double a, b;

public Harmonic(double newA, double newB){
a = newA;
b = newB;
}
public double value(double X){
return a*sin(X) + b*cos(X);
}
}

1.3| Harmonic h = new Harmonic(a, b);
Function f = new Function(h, 0,0001);
System.out.print(f.max(-1,5; 1,5));

```
public class Knight extends ChessPiece {
    private Rectangle cap;
    public Knight (int size){
        super(size);
        cap = new Rectangle(0, 1/6, 2/3, 1/3);
    }
    public void drawCap(Graphics g){
        g.drawOval(cap.x, cap.y, field.width, field.height);
    }
}
```

3] ①

```
for (row = 0; row < grid.length; row++){
    for(col = 0; col < grid[0].length; col++){
        if(sumG(row,col) == 3)
            grid[row][col] = true;
        else{
            if(sumG(row,col) == 2 && grid[row][col] == true)
                grid[row][col] = true;
            else
                grid[row][col] = false;
        }
}
```

5

```
for( int row=0, row< grid.length; row++) {
    for(int col=0; col<grid[0].length, col++){
        if( grid[row][col])
            g.fillRect(col*4, row*4, 4, 4);
        else
            g.drawRect(col*4, row*4, 4,4);
    }
}
```