Name and, if possible, ID#:

AMERICAN UNIVERSITY OF ARMENIA

College of Science and Engineering

COMP120 Introduction to Object-Oriented Programming

FINAL EXAM

9/15

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 derivative at the specified point ***x*** using the approximation:

$$f'(x) \approx \frac{f(x+dx) - f(x-dx)}{(2*dx)}$$

```
public class Function {

      private Valuable f;
      private double dx;

      public Function(Valuable newValuable, double newDX) {
            //TO BE IMPLEMENTED
      }

      public double derivative(double x) {
            //TO BE IMPLEMENTED
      }
```

1.2 Implement an expression

exp(-a * (x – c)²)

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

1.3 In a separate ***public static void main(String args[])*** write a code that inputs two double values, creates an object of type ***Gauss*** and, using the class ***Function***, prints the value of its derivative at the ***x = 1.0*** point:

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

      //TO BE COMPLETED

}
```

2

see LM

Use the backside, if needed

OOP.FT.180515.H101

```
1    public Function (Valuable new Valueable, double new DX){
            f = new Valuuble;
            X = newDX. ;

     }

     public  double  derivative (double x) {
          function derivative = new function derivative();
     derivative f(x) derivative dX = (Value(X+dX)  - value(X-dX)) / (2 * dX);
     }
1. 2
   Public class  Gauss {
          private double  a, c;
     public Gauss(double newA, double newC).
               a = newA;
               c = newC;
     public double  gGauss(double x).
          Gauss g = new Gauss();

          e.Math.Pow(-g.a * (x-g.c) * (x-g.c)).
1. 3 Public static  void main(String args[]) {
        Scanner  input = new Scanner(System.in);
          double a = input.nextDouble(), c = input nextDouble();
             Gauss  object = new Gauss();
               Function  = new Function();
                  derivative(1.0);

     }
```

```
2. Private   Rectangle cap;

 public class  Rook. extends  ChessPiece {
  public Rook() {
   cap =new  Rectangle(size/6, -size/6, 2*size/3, sine/3).
 }

  public void drawCap(Grapics g) { g. setcolor(Black);
        g. drawRect( cap.x , cap. y , cap.width , cap.height);

  }

3. public Boolean tick() {
     for(int i=0; i<=100; i++)
        for (int j=0; j<=100; j++)      if grid[i][j]==true {
     if (sum (i,j) <2  || sum(i,j)>3)
            grid[i][j]= false;
     if (sum(i,j)=2  || sum(i,j)=3)
              grid[i][j]= true;
     }
     else   if(sum(i,j) = 3
              grid[i][j]= true;
               return true;

 public void   snapshot ( Graphics g) { for(int i= 0; i<=100, i++){
                                         for(int j=0, j<=100, j++) {
    g. setcolorwhite();
       if (grid[i][j]=true)
    g. drawRect(40,0,400,4,4).
         else (if grid [i][j]= false)
          {
          g. setcolor Black();
          ~~g.drawRect(400,400,4,4);~~
         g.fillRect. ;

   }
  }
```