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 integral in the specified range from x_1 to x_2 using the approximation:

$$\int_{x1}^{x2} f(x)dx \approx \frac{x_2 - x_1}{6}\left(f(x_1) + 4f\left(\frac{x_1 + x_2}{2}\right) + f(x_2)\right)$$

```
public class Function {

       private Valuable f;
       private double dx;

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

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

1.2 Implement an expression

$$\sqrt{x^2 + a} + \sqrt{x^2 + b}$$

as a ***public class Roots*** that implements the interface ***Valuable*** and encapsulates double parameters ***a*** and ***b***. The parameters are initialized by the two-argument constructor ***public Roots(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 ***Roots*** and, using the class ***Function***, prints the value of its integral from $x_1 = 1.0$ to $x_1 = 2.0$:

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

       //TO BE COMPLETED

}
```

2

See N4

Use the backside, if needed

OOP.FT.180515.H098

a Metsabyan

```
1
ublie Function(Valuable newValuable, double newDX){
            f = newValuable;
            dx = newDX;
        }
public double integral(double x1, double x2) {
              Function intg = new Function intg();
 intg.f*intg.dx
~~double result~~ = (x2-x1)/6*(value(x1)+4*value((x1+x2)/2)+value(x2))
              ~~return result;~~

       }
1.2
Public class Roots {
    Private double a, b;
    public Roots(double newA, double newB) {
              a = newA;
              b = newB;
        }
    ~~public double sqr~~ public double sroot(double x) {
              Roots r = new Roots();
               sqrt(x*x + r.a) + sqrt(x*x + r.b); }

1.3
public static void main(String args[]) {
          scanner input = new Scanner(system.in);
      double a = input.nextDouble(); b = input.nextDouble();
         Roots obj = new Roots();
           Function rd = new Function();
               integral(1.0, 2.0);


}
```

```
                                                    {

   public Bishop(int size){
        field=new Rectange(size*2/3, size*1/3);
        ~~inner = new Oval~~ }
  @override
  public void drawCap(Graphics g){
           g.setcolor(black);
       g.drawOval(size/6, 2*size/3, size/3);
   }
                   -size/6
```

problem3

```
  Public boolean tick() {
      ~~i~~for(int i= 0; i< 100; i++)
         for(int j=0; j<100; j++){if[grid[i][j] == true) {
     if (sum9(i,j)< 2 || sum9(i,j) >3 )
           grid[i][j]= false;
     if (sum9(i,j)== 2 && sum9(i,j)== 3 )
           grid[i][j] =true;
      }
       else ~~[illegible]~~{
          if (sum9(i,j) == 3)
              grid[i][j]= true;
                   } return ~~grid~~ false;
                  }
public ~~boolean~~ void snapshot(Graphics g){
       g.setColor(white);
       g.drawRect(0, 0, 100,100);
       g.fillRect(0,0, 100, 100);
       g.setColor(black)
~~[illegible]~~ for(int i=0; i<100; i++)
       for(ind j=0; j<100; j++) {
```

```
for (int l =0; l<100; l += 4)
    for (int r =0; r <100; l+=4)
      g.drawRect(l, r, 4, 4);
    if ( grid [i][j] == true )
         g.fillRect (l, r, 4, 4);

}
```