

CSCI 1101: LAB NO. 2

SOLUTIONS (Note: Only solution outlines for relevant exercises are given here).

```
public class Bug
{
    private int pos;
    private int dir;

    public Bug(int p, int d)
    {
        pos = p;
        dir = d;
    }
    public int getPosition()
    {
        return pos;
    }
    public int getDirection()
    {
        return dir;
    }
    public void setPosition(int p)
    {
        pos = p;
    }
    public void setDirection(int d)
    {
        dir = d;
    }
    public void move()
    {
        if (dir==0)
            if (pos>0)
                pos=pos-1;
            else
            {
                dir=1;
                pos=pos+1;
            }
        else if (dir==1)
            if (pos<50)
                pos=pos+1;
            else
            {
                dir=0;
                pos=pos-1;
            }
    }
}
```

```

    }
    public String toString()
    {
        String result="";
        result+="Position: " + pos + "\t";
        if (dir == 0)
            result+="Direction: Left";
        else
            result+="Direction: Right";
        return result;
    }

    public static void main(String[] args)
    {
        //randomly set the initial position (between 0 and 50)
        and direction (between 0 and 1)
        int initialPos = (int)(Math.random()*51);
        int initialDir = (int)(Math.random()*2);
        Bug bugsy = new Bug(initialPos, initialDir);
        System.out.println(bugsy);

        //display the line with the bug
        for(int i=0; i<=50;i++)
        {
            if (i==bugsy.getPosition())
                System.out.print("X");
            else
                System.out.print("-");
        }
        System.out.println();

        //make the bug move a random number of times
        int moves = (int)(Math.random()*51);
        System.out.println("Moves: " + moves);
        for(int i=1;i<=moves;i++)
            bugsy.move();
        System.out.println(bugsy);

        //display the line with the bug
        for(int i=0; i<=50;i++)
        {
            if (i==bugsy.getPosition())
                System.out.print("X");
            else
                System.out.print("-");
        }

        System.out.println();
    }

```

```

    }

}

public class Point
{
    private int xpos;
    private int ypos;

    public Point(int x, int y)
    {
        xpos=x;
        ypos=y;
    }
    public void setX(int x)
    {
        xpos=x;
    }
    public void setY(int y)
    {
        ypos=y;
    }
    public int getX()
    {
        return xpos;
    }
    public int getY()
    {
        return ypos;
    }
    public String toString()
    {
        return "["+xpos+","+ypos+"]";
    }
    public boolean isHigher(Point other)
    {
        return (ypos<other.getY());
    }
    public double findDist(Point other)
    {
        double answer;
        answer = (other.getX()-xpos)*(other.getX()-xpos)+
(other.getY()-ypos)*(other.getY()-ypos);
        answer = Math.sqrt(answer);
        return answer;
    }
}

```

```

import java.util.Scanner;
public class PointDemo1
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the x and y coordinates of
point1: " );
        Point p1 = new Point(input.nextInt(),
input.nextInt());
        System.out.print("Enter the x and y coordinates of
point2: " );
        Point p2 = new Point(input.nextInt(),
input.nextInt());
        System.out.print("Enter the x and y coordinates of
point3: ");
        Point p3 = new Point(input.nextInt(),
input.nextInt());
        System.out.print("Enter the x and y coordinates of
point4: ");
        Point p4 = new Point(input.nextInt(),
input.nextInt());

        if (p1.isHigher(p2) &&
p1.isHigher(p3)&&p1.isHigher(p4))
            System.out.println(p1 + " is the highest
point");
        else if
(p2.isHigher(p1)&&p2.isHigher(p3)&&p2.isHigher(p4))
            System.out.println(p2 + " is the highest
point");
        else if
(p3.isHigher(p1)&&p3.isHigher(p2)&&p3.isHigher(p4))
            System.out.println(p3 + " is the highest
point");

        double length1 = p1.findDist(p2);
        double length2 = p3.findDist(p4);
        System.out.println("The distance between " + p1 + "
and " + p2 + " is " + length1);
        System.out.println("The distance between " + p3 + "
and " + p4 + " is " + length2);
        if (length1>length2)
            System.out.println(p1 + "-->" + p2 + " is
longer than " + p3 + "-->" + p4);
        else if (length2>length1)
            System.out.println(p3 + "-->" + p4 + " is
longer than " + p1 + "-->" + p2);
        else

```

```

        System.out.println("Both distances are the
same");
    }
}

public class Rectangle2D
{
    private double xpos, ypos, width, height;

    public Rectangle2D()
    {
        xpos = 0;
        ypos = 0;
        width = 0;
        height = 0;
    }

    public Rectangle2D(double xpos, double ypos, double width,
double height)
    {
        this.xpos = xpos;
        this.ypos = ypos;
        this.width = width;
        this.height = height;
    }

    public void setX(double xpos)
    {
        this.xpos = xpos;
    }

    public void setY(double ypos)
    {
        this.ypos = ypos;
    }

    public void setWidth(double width)
    {
        this.width = width;
    }

    public void setHeight(double height)
    {
        this.height = height;
    }

    public double getX()
    {
        return xpos;
    }

    public double getY()
    {
        return ypos;
    }

    public double getWidth()

```

```

    {
        return width;
    }
    public double getHeight()
    {
        return height;
    }

    public double findPerimeter()
    {
        return 2*height+2*width;
    }
    public double findArea()
    {
        return height*width;
    }
    public String toString()
    {
        String result = "["+ xpos + "," + ypos + "],";
        result = result + "width=" + width + "," + "height="
+ height + "];"
        return result;
    }
    public boolean contains(double px, double py)
    {
        return px >=xpos && px<=xpos+width && py>=ypos &&
py<=ypos+height;
    }
    public boolean contains(Rectangle2D other)
    {
        double p1x = other.getX();
        double p1y = other.getY();
        double p2x = other.getX()+other.getWidth();
        double p2y = other.getY()+other.getHeight();

        if (this.contains(p1x,
p1y)&&this.contains(p1x,p2y)&&this.contains(p2x,p1y)&&this.contains(p2
x,p2y))
            return true;
        else
            return false;
    }
}

```

```

import java.util.Scanner;
public class Rectangle2DDemo
{
    public static void main(String[] args)

```

```

    {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the xpos, ypos, width and
height of the rectangle: " );
        Rectangle2D r1 = new Rectangle2D(input.nextDouble(),
input.nextDouble(), input.nextDouble(), input.nextDouble());

        System.out.println("The perimeter of the rectangle is
" + r1.findPerimeter());
        System.out.println("The area of the rectangle is " +
r1.findArea());
        if (r1.contains(3,3))
            System.out.println("Rectangle " + r1 + "
contains point [3,3]");
        else
            System.out.println("Rectangle " + r1 + " does
not contain point [3,3]");
        Rectangle2D r2 = new Rectangle2D(4,5,10.5,3.2);
        if (r1.contains(r2))
            System.out.println(r1 + " contains " + r2);
        else
            System.out.println(r1 + " does not contain
Rectangle " + r2);
    }
}

```

```

public class Stock
{
    private String symbol;
    private double price;
    private int shares;

    public Stock(String symbol, double price, int shares)
    {
        this.symbol = symbol;
        this.price = price;
        this.shares = shares;
    }
    public void setSymbol(String symbol)
    {
        this.symbol = symbol;
    }
    public void setPrice(double price)
    {
        this.price = price;
    }
    public void setShares(int shares)
    {
        this.shares = shares;
    }
}

```

```

    }
    public String getSymbol()
    {
        return symbol;
    }
    public double getPrice()
    {
        return price;
    }
    public int getShares()
    {
        return shares;
    }
    public String toString()
    {
        String result = "Symbol: " + symbol + "\n";
        result+= "Price: " + price + "\n";
        result+="Shares: " + shares + "\n";
        return result;
    }
    public int compare(Stock s)
    {
        double value1 = price*shares;
        double value2 = s.getPrice()*s.getShares();
        if (value1>value2)
            return 1;
        else if (value2>value1)
            return -1;
        else
            return 0;
    }
}

import java.util.Scanner;
public class StockDemo
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        String sym1, sym2;
        double prc1, prc2;
        int sh1, sh2;
        //get the values for two stocks
        System.out.print("Enter the symbols for the two
stocks: ");
        sym1 = keyboard.next();
        sym2 = keyboard.next();
        System.out.print("Enter their prices: ");
        prc1 = keyboard.nextDouble();

```



```

        prc2 = keyboard.nextDouble();
        System.out.print("Enter the number of shares for the
two stocks: ");
        sh1 = keyboard.nextInt();
        sh2 = keyboard.nextInt();

        //create the first Stock
        Stock s1 = new Stock(sym1,prc1,sh1);

        //create the second Stock
        Stock s2 = new Stock(sym2,prc2,sh2);

        System.out.println("I have the following stocks: ");
        System.out.println(s1);
        System.out.println(s2);
        int c = s1.compare(s2);
        if (c==1)
            System.out.println("The value of " +
s1.getSymbol() + " is higher than " + s2.getSymbol());
        else if (c==-1)
            System.out.println("The value of " +
s2.getSymbol() + " is higher than " + s1.getSymbol());
        else
            System.out.println("The values of " +
s1.getSymbol() + " and " + s2.getSymbol() + " are the same.");
        System.out.println("The total value in my portfolio is
$ : " + (s1.getPrice()*s1.getShares() +
s2.getPrice()*s2.getShares()));
    }
}

```