CS 342 Report for (Lab 06

Lab6.class

```
//Programmer :Fayaz Khan
//Assignment :Lab 5
//Date:       September 29, 2015
//Description:Program uses test drivers to test methods in Enemy class.

public class Lab6
{

    public static void main(String[] args)
    {

        Enemy enemy1 = new Enemy();
        Enemy enemy2 = new Enemy(5,2);
        Enemy enemy3 = new Enemy(85,3);


        System.out.println("enemy1: " + enemy1.toString());
        System.out.println("enemy2: " + enemy2.toString());
        System.out.println("enemy3: " + enemy3.toString());

        enemy1.hit();
        enemy2.hit();
        enemy3.hit();

        System.out.println("enemy1, after hit: " + enemy1.toString());
        System.out.println("enemy2, after hit: " + enemy2.toString());
        System.out.println("enemy3, after hit: " + enemy3.toString());

        enemy1.inactive();

        System.out.println("enemy1, strength: " + enemy1.strength());
        System.out.println("enemy1 inactive, danger: " + enemy1.danger(1));

        enemy2.active();

        System.out.println("enemy2, strength: " + enemy2.strength());
        System.out.println("enemy2 level 2 active, danger: " + enemy2.danger(12));

        System.out.println("enemy3,strength:" + enemy3.strength() + " danger: " + enemy3.danger(9));

        System.out.println("enemy1 after active method, danger: " + enemy1.danger(1));

        if(enemy1.battle(enemy2))
            System.out.println("Enemy1 is stronger than Enemy2");
        else
            System.out.println("Enemy2 is stronger than Enemy1");

        if(enemy3.battle(enemy2))
            System.out.println("Enemy2 is stronger than Enemy3");
        else
            System.out.println("Enemy3 is stronger than Enemy2");

    }

}
```

Enemy.class

```
//Programmer : Fayaz Khan
//Assignment : Lab 5
//Date:        September 29, 2015
//Description: Includes methods to construct objects and utilizing overloading methods.Includes
//             data members how strong enemies are,how much strength they lose per hit, the
//             threshold strength level for enemies being dangerous,their vertical levels,and
//             whether or not they are active.Methods include for Enemy default constructor,
//             initializer constructor,modifier that processes the effects of an enemy suffering
//             a single hit,two modifiers renders them inactive or active. Enemy class also
//             contains accessors  to get the current strength of an enemy,get the danger the
//             enemy poses to the player located at a given vertical level, and allows two enemies'
//             strengths to be compared.

public class Enemy
{
    private static final int THRESHOLD = 15;    //the minimum for an enemy to be a threat
    private static final int FULLSTRENGTH = 100;//default starting for each enemy
    private final int damage;                   //Amount of strength subtracted after each hit
    private static boolean active;              //true if active, false otherwise
    private int strength;                       //Enemy strength starts at 100 and ends at 0
    private boolean threat;                     //strength >= 15 threat is true,false otherwise
    private int verticalLevel;                  //location of enemies

    public Enemy ()
    // POST: default Enemy object strength is set to full, DAMAGE is set to ten, threat == true
    //       place on vertical level 1 and is active call is made to the other Enemy constructor
    {
        this(10,1);
    }

    public Enemy(int damage,int verticalLevel)
    // PRE:   damage is >= 0
    //        verticalLevel >= 1
    // POST:  a Enemy object is created with with full strength, threat == true and active == true.
    //        (this.damage) == damage and (this.verticalLevel) == verticalLevel.
    {
        strength = FULLSTRENGTH;
        this.damage = damage;
        threat = true;
        this.verticalLevel = verticalLevel;
        active = true;
    }

    public void hit()
    // POST: when an enemy is hit subtracts damage from strength and then updates strength value
    {
        strength = strength - damage;
    }

    public int strength()
    // POST: FCTVAL == current strength of enemy
    {
        return strength;
    }

    public void inactive()
    // POST: makes enemy inactive by setting active == false
    {
        active = false;
    }

    public void active()
    // POST: makes enemy active by setting active == true
    {
        active = true;
    }
```

```
    public int danger(int playerLevel)
    // PRE:  playerLevel >= 1
    // POST: FCTVAL == active == false returns 0
    //        FCTVAL == strength > 15 returns strength minus vertical level; otherwise,
    //        FCTVAL == 0;
    {
        int checkDanger = 0;                  //to make sure the danger not returned is less than 0

        if(active == false)
            return 0;

        if(strength > THRESHOLD)
        {
            checkDanger = strength-(Math.abs(playerLevel-verticalLevel));

            if(checkDanger >= 0)
                return checkDanger;
            else
                return 0;
        }
        else
            return 0;
    }

    public boolean battle(Enemy enemy)
    // PRE:  enemy is initialized
    // POST: FCTVAL == true when this.strength > enemy.strength;
    //        FCTVAL == false otherwise
    {
        if(this.strength > enemy.strength)
            return true;
        else
            return false;
    }

    public String toString()
    // POST: FCTVAL == returns all information about object
    {
        return "Enemy Strength: " + strength + " Damage: " + Damage + " level: " + verticalLevel;
    }

}
```

Sample Run

```
enemy1: Enemy Strength: 100 Damage: 10 level: 1
enemy2: Enemy Strength: 100 Damage: 5 level: 2
enemy3: Enemy Strength: 100 Damage: 85 level: 3
enemy1, after hit: Enemy Strength: 90 Damage: 10 level: 1
enemy2, after hit: Enemy Strength: 95 Damage: 5 level: 2
enemy3, after hit: Enemy Strength: 15 Damage: 85 level: 3
enemy1, strength: 90
enemy1 inactive, danger: 0
enemy2, strength: 95
enemy2 level 2 active, danger: 85
enemy3,strength:15 danger: 0
enemy1 after active method, danger: 90
Enemy2 is stronger than Enemy1
Enemy2 is stronger than Enemy3
```