Hero Testing

```
public class TestGame extends TestCase {
public void testHero() {
Hero h = new Hero();
h.setName(“Tom”);
assertEquals((“Tom”,g.getName());
}
}
```

Second round of testing

```
public class TestGame extends TestCase {
public void testName() {
Hero h = new Hero();
h.setName(“Tom”);
assertEquals(“Tom”, h.getName());
}
public void testElement() {
Hero h = new Hero();
h.setElement(“Fire”);
assertEquals(“Fire”, h.getElement());
}
}

public void testGender() {
Hero h = new Hero();
h.setName(“Male”);
assertEquals(“Male”, h.getGender());
}

public void testHP() {
Hero h = new Hero();
h.setHP(100);
assertEquals(100, h.getHP());
}
public void TestDamange(){
Hero H = new Hero();
h.setHP(100);
h.takeDMG(25);
assertEquals(75, h.getHP());
}
public void TestHeal(){
Hero H = new Hero();
h.setHP(100);
h.takeDMG(25);
h.heal(10);
assertEquals(85, h.getHP());
}
```

Weapon Testing

```
public void testUpgrade(){
Weapon one = new Weapon();
one.setUpgrade(1);
one.upgrade();
int testAtk = one.getWpatk();
assertEquals(testAtk,17);
}
Public void testWepCheck(){
Weapon two = new Weapon();
two.setUpgrade(999);
two.wepCheck();
}
```

Battle Testing

```
Public void testCharCompare(){
        String a = “tom”;
        String e = “tom”;
        Double compared = 0.0;
        Battle test = new Battle();
        compared = Test.charCompare(a,e);
        assertEquals(1,compared);
}

Public void testCharCompare2(){
        String a = “tom”;
        String e = “toms”;
        Double compared = 0.0;
        Battle test = new Battle();
        compared = Test.charCompare(a,e);
        assertEquals(.75,compared);
}
```

Monster Testing

```
public void testHP() {
        monster m = new Monster();
        m.setHP(100);
        assertEquals(100, h.getHP());
}

public void TestDamange(){
```

```
        monster m = new Monster();
        m.setHP(100);
        m.takeDMG(25);
        assertEquals(75, h.getHP());
}
```

```
package Team4;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;


public class FrameTest {

    public FrameTest() {
    }
    Frame ne = new Frame();

    @BeforeClass
    public static void setUpClass() throws Exception {
    }

    @AfterClass
    public static void tearDownClass() throws Exception {
    }
    @Test
    public void testgetIndex(){
        System.out.println("DO DAMAGE");

       if(ne.getIndex() < 15 && ne.getIndex() >= 0)
       {
           System.out.println("SUCCESS");
       }

}
       @Test
    public void testActualWord(){
        System.out.println("ACTUAL WORD");

      int n = ne.getIndex();
      boolean expResult = true;
     boolean m = ne.wordList[n].equals(ne.actualWord());

     assertEquals(expResult,m);
       }
}
```

```
package Team4;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class MonsterTest {

      Monster instance = new Monster();

    public MonsterTest() {
    }

    @BeforeClass
    public static void setUpClass() throws Exception {
         Monster instance = new Monster();
    }

    @AfterClass
    public static void tearDownClass() throws Exception {
    }
   @Test
    public void testInitialization(){
        System.out.println("Constructor");
        int expResult = 30;
        int res = 5;
        assertEquals(expResult, instance.getHealth());
        assertEquals(res, instance.doDamage());

    }
```

```
    @Test
    public void testDoDamge(){
        System.out.println("Constructor");
       int expResult = 5;
       assertEquals(expResult, instance.doDamage());

    }
    @Test
    public void testGetHealth() {
        System.out.println("getHealth");
        int expResult = 30;
        int result = instance.getHealth();
        assertEquals(expResult, result);


    }
@Test
    public void testTakeDamage() {
        System.out.println("getDamage");
        instance.takeDamage(instance.doDamage());
        assertEquals(25,instance.getHealth());


    }
    @Test
    public void testSetHealth() {
        System.out.println("setHealth");
        int expResult = 500;
        instance.setHealth(500);
        assertEquals(expResult, instance.getHealth());
    }

    @Test
    public void testDeadCheck() {
        System.out.println("Die");
        boolean expResult = false;
        boolean result = instance.deadCheck();
        assertEquals(expResult, result);


    }



}
```