MonsterPrime.java MonsterTester.java

Write the MonsterPrime class based on the following MonsterTester class:

```
import static java.lang.System.*;

public class MonsterTester
{
  public static void main( String args[] )
  {
                MonsterPrime zero = new MonsterPrime();
                MonsterPrime one = new MonsterPrime(8);
                MonsterPrime sue = new MonsterPrime(9, 4);
                MonsterPrime harry = new MonsterPrime(1, 2, 3);
                out.println("\nzero Monster :: " + zero);
                out.println("\none Monster :: " + one);
                out.println("\nsue Monster :: " + sue);
                out.println("\nharry Monster :: " + harry);

                out.println("\nchanging harry's properties ");
                harry.setHeight(7);
                harry.setWeight(6);
                harry.setAge(5);
                out.println("\nharry Monster :: " + harry);

                out.println("\ncloning harry");
                sue = (MonsterPrime)harry.clone();
                out.println("\nsue Monster :: " + sue);

                MonsterPrime mOne = new MonsterPrime(33,33,11);
                MonsterPrime mTwo = new MonsterPrime(55,33,11);

                out.println("\nMonster 1 :: " + mOne);
                out.println("\nMonster 2 :: " + mTwo);

                out.print("\nmOne.equals(mTwo) == ");
                out.println(mOne.equals(mTwo));

                out.print("\nmOne.compareTo(mTwo) == ");
                out.println(mOne.compareTo(mTwo));
                out.print("\nmTwo.compareTo(mOne) == ");
                out.println(mTwo.compareTo(mOne));
        }
}
```

MonsterPrime will use the Comparable interface. When comparing return either a -1, 0, or 1. When comparing, the criteria will be 1st – height, 2nd – weight, and 3rd – age. When writing the constructors, the order of the paramenter will go height, weight, and age. When writing the toString, the order should go height, weight, and age.

OUTPUT

```
zero Monster :: 0 0 0

one Monster :: 8 0 0

sue Monster :: 9 4 0

harry Monster :: 1 2 3

changing harry's properties

harry Monster :: 7 6 5

cloning harry

sue Monster :: 7 6 5

Monster 1 :: 33 33 11

Monster 2 :: 55 33 11

mOne.equals(mTwo) == false

mOne.compareTo(mTwo) == -1

mTwo.compareTo(mOne) == 1
```