```java
package edu.seaaddicts.brockbutler.maps;

/**
 * Position.java
 * Brock Butler
 * Type for holding Position node
 * portion of Brock Butler.
 * Created by Thomas Nelson 2013-03-05
 * Copyright (c) 2013 Sea Addicts. All rights reserved.
 */

import android.util.Log;

public class Position implements Comparable<Object> {

  /**
   * Class variable for the POSITION class. All are public
   * to avoid using get/set variables to increase performance
   */
  public int      xPosition;
  public int      yPosition;
  public double   fScore;
  public double   gScore;
  public double   hScore;
  public String   nodeNumber;
  public String   nodeName;
  public boolean  visited;
    public Position from;
    public Position accesible[];
    public Position nonaccesible[];

  /**
   * Constructor methods for no arguments
   */
  public Position ( ) {
    xPosition = 0;
    yPosition = 0;

    nodeNumber = "";
    nodeName   = "";

    fScore = Double.MAX_VALUE;
        gScore = Double.MAX_VALUE;
        hScore = -1;

        visited = false;
        from    = null;
  }

  /**
   * Constructor with coordinates set
   * @param inputX
   * @param inputY
   */
  public Position (int inputX, int inputY) {
    xPosition = inputX;
    yPosition = inputY;

    nodeNumber = "";
    nodeName   = "";

    fScore = Double.MAX_VALUE;
    gScore = Double.MAX_VALUE;
    hScore = Double.MAX_VALUE;

    visited = false;
        from    = null;
  }

  /**
   * Constructor with all position information set
```

```
     * @param inputX
     * @param inputY
     * @param inputName
     * @param inputNumber
     */
    public Position (int inputX, int inputY, String inputName, String inputNumber) {
      xPosition = inputX;
      yPosition = inputY;

      nodeNumber = inputNumber;
      nodeName   = inputName;

      fScore = Double.MAX_VALUE;
      gScore = Double.MAX_VALUE;
      hScore = Double.MAX_VALUE;

      visited = false;
          from    = null;
    }

    /**
     * Set coordinates
     * @param inputX
     * @param inputY
     */
    public void setCoordinates (int inputX, int inputY) {
      xPosition  = inputX;
      yPosition  = inputY;
    }

    /**
     * Set position number
     * @param inputNumber
     */
    public void setNumber (String inputNumber) {
      nodeNumber  = inputNumber;
    }

    /**
     * Set position description
     * @param inputName
     */
    public void setName (String inputName) {
      nodeName  = inputName;
    }

    /**
     * get x coordinate
     * @return
     */
    public int getX ( ) {
      return xPosition;
    }

    /**
     * get y coordinate
     * @return
     */
    public int getY ( ) {
      return yPosition;
    }

    /**
     * get node numner
     * @return
     */
    public String getNumber ( ) {
      return nodeNumber;
    }

    /**
```

```
     * Get node name
     * @return
     */
    public String getName ( ) {
      return nodeName;
    }

    /**
     * Compares this node to another
     * @param node
     * @return
     */
    public boolean compare (Position node) {
      if(this.xPosition == node.xPosition && this.yPosition == node.yPosition && this.nodeNumber.equals(node.nodeNumber) && this.nodeName.equals(node.nodeName))
        return true;
      return false;
    }

    /**
     * Not Used but required???
     */
    public int compareTo (Object node) {
      Position temp = (Position)node;
          return (int)(fScore - temp.fScore);
    }


    /**
     * Testing methods for the POSITION class. These methods are provided
     * for testing and debugging purposes capable of printing variables to the log
     */
    public void printCoordinates ( ) {
      Log.d("POSITION CLASS", "Coordinates: (" + xPosition + "," + yPosition + ")");
    }

    public void printNumber ( ) {
      Log.d("POSITION CLASS", "Node Number: " + nodeNumber);
    }

    public void printName ( ) {
      Log.d("POSITION CLASS", "Node Name: " + nodeName);
    }
  }
```