AWT CUBICCURVE2D CLASS

http://www.tutorialspoint.com/awt/awt_cubiccurve2d_class.htm

Introduction

The CubicCurve2D class states a cubic parametric curve segment in *x*, *y* coordinate space.

Class declaration

Following is the declaration for **java.awt.geom.CubicCurve2D**class:

```
public abstract class CubicCurve2D
   extends Object
      implements Shape, Cloneable
```

Class constructors

S.N.	Constructor & Description
1	**protected CubicCurve2D** This is an abstract class that cannot be instantiated directly.

Class methods

S.N.	Method & Description
1	**Object clone** Creates a new object of the same class as this object.
2	**boolean contains***doublex, doubley* Tests if the specified coordinates are inside the boundary of the Shape.
3	**boolean contains***doublex, doubley, doublew, doubleh* Tests if the interior of the Shape entirely contains the specified rectangular area.
4	**boolean contains***Point2Dp* Tests if a specified Point2D is inside the boundary of the Shape.
5	**boolean contains***Rectangle2Dr* Tests if the interior of the Shape entirely contains the specified Rectangle2D.
6	**Rectangle getBounds**

Returns an integer Rectangle that completely encloses the Shape.

7

abstract Point2D getCtrlP1

Returns the first control point.

8

abstract Point2D getCtrlP2

Returns the second control point.

9

abstract double getCtrlX1

Returns the X coordinate of the first control point in double precision.

10

abstract double getCtrlX2

Returns the X coordinate of the second control point in double precision.

11

abstract double getCtrlY1

Returns the Y coordinate of the first control point in double precision.

12

abstract double getCtrlY2

Returns the Y coordinate of the second control point in double precision.

13

double getFlatness

Returns the flatness of this curve.

14

static double getFlatness*double[]coords, intoffset*

Returns the flatness of the cubic curve specified by the control points stored in the indicated array at the indicated index.

15

static double getFlatness
doublex1, doubley1, doublectrlx1, doublectrly1, doublectrlx2, doublectrly2, doublex2, doubley2

Returns the flatness of the cubic curve specified by the indicated control points.

16

double getFlatnessSq

Returns the square of the flatness of this curve.

17

static double getFlatnessSq*double[]coords, intoffset*

Returns the square of the flatness of the cubic curve specified by the control points stored in the indicated array at the indicated index.

18

static double getFlatnessSq
doublex1, doubley1, doublectrlx1, doublectrly1, doublectrlx2, doublectrly2, doublex2, doubley2

Returns the square of the flatness of the cubic curve specified by the indicated control points.

19

abstract Point2D getP1

Returns the start point.

20

abstract Point2D getP2

Returns the end point.

21

PathIterator getPathIterator*AffineTransformat*

Returns an iteration object that defines the boundary of the shape.

22

PathIterator getPathIterator*AffineTransformat, doubleflatness*

Return an iteration object that defines the boundary of the flattened shape.

23

abstract double getX1

Returns the X coordinate of the start point in double precision.

24

abstract double getX2

Returns the X coordinate of the end point in double precision.

25

abstract double getY1

Returns the Y coordinate of the start point in double precision.

26

abstract double getY2

Returns the Y coordinate of the end point in double precision.

27

boolean intersects*doublex, doubley, doublew, doubleh*

Tests if the interior of the Shape intersects the interior of a specified rectangular area.

28

boolean intersects*Rectangle2Dr*

Tests if the interior of the Shape intersects the interior of a specified Rectangle2D.

29

void setCurve*CubicCurve2Dc*

Sets the location of the end points and control points of this curve to the same as those in the specified CubicCurve2D.

30

void setCurve*double[]coords, intoffset*

Sets the location of the end points and control points of this curve to the double coordinates at the specified offset in the specified array.

31 **abstract void setCurve**
doublex1, doubley1, doublectrlx1, doublectrly1, doublectrlx2, doublectrly2, doublex2, doubley2

Sets the location of the end points and control points of this curve to the specified double coordinates.

32

void setCurve*Point2D[]pts, intoffset*

Sets the location of the end points and control points of this curve to the coordinates of the Point2D objects at the specified offset in the specified array.

33

void setCurve*Point2Dp1, Point2Dcp1, Point2Dcp2, Point2Dp2*

Sets the location of the end points and control points of this curve to the specified Point2D coordinates.

34

static int solveCubic*double[]eqn*

Solves the cubic whose coefficients are in the eqn array and places the non-complex roots back into the same array, returning the number of roots.

35

static int solveCubic*double[]eqn, double[]res*

Solve the cubic whose coefficients are in the eqn array and place the non-complex roots into the res array, returning the number of roots.

36

void subdivide*CubicCurve2Dleft, CubicCurve2Dright*

Subdivides this cubic curve and stores the resulting two subdivided curves into the left and right curve parameters.

37

static void subdivide*CubicCurve2Dsrc, CubicCurve2Dleft, CubicCurve2Dright*

Subdivides the cubic curve specified by the src parameter and stores the resulting two subdivided curves into the left and right curve parameters.

38

static void subdivide*double[]src, intsrcoff, double[]left, intleftoff, double[]right, intrightoff*

Subdivides the cubic curve specified by the coordinates stored in the src array at indices srcoff through $srcoff + 7$ and stores the resulting two subdivided curves into the two result arrays at the corresponding indices.

Methods inherited

This class inherits methods from the following classes:

- java.lang.Object

CubicCurve2D Example

Create the following java program using any editor of your choice in say **D:/ > AWT > com > tutorialspoint > gui >**

AWTGraphicsDemo.java

```
package com.tutorialspoint.gui;

import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class AWTGraphicsDemo extends Frame {

   public AWTGraphicsDemo(){
      super("Java AWT Examples");
      prepareGUI();
   }

   public static void main(String[] args){
      AWTGraphicsDemo  awtGraphicsDemo = new AWTGraphicsDemo();
      awtGraphicsDemo.setVisible(true);
   }

   private void prepareGUI(){
      setSize(400,400);
      addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
         }
      });
   }

   @Override
   public void paint(Graphics g) {
      CubicCurve2D shape = new CubicCurve2D.Float();
      shape.setCurve(250F,250F,20F,90F,140F,100F,350F,330F);
      Graphics2D g2 = (Graphics2D) g;
      g2.draw (shape);
      Font font = new Font("Serif", Font.PLAIN, 24);
      g2.setFont(font);
      g.drawString("Welcome to TutorialsPoint", 50, 70);
      g2.drawString("CubicCurve2D.Curve", 100, 120);
   }
}
```

Compile the program using command prompt. Go to **D:/ > AWT**

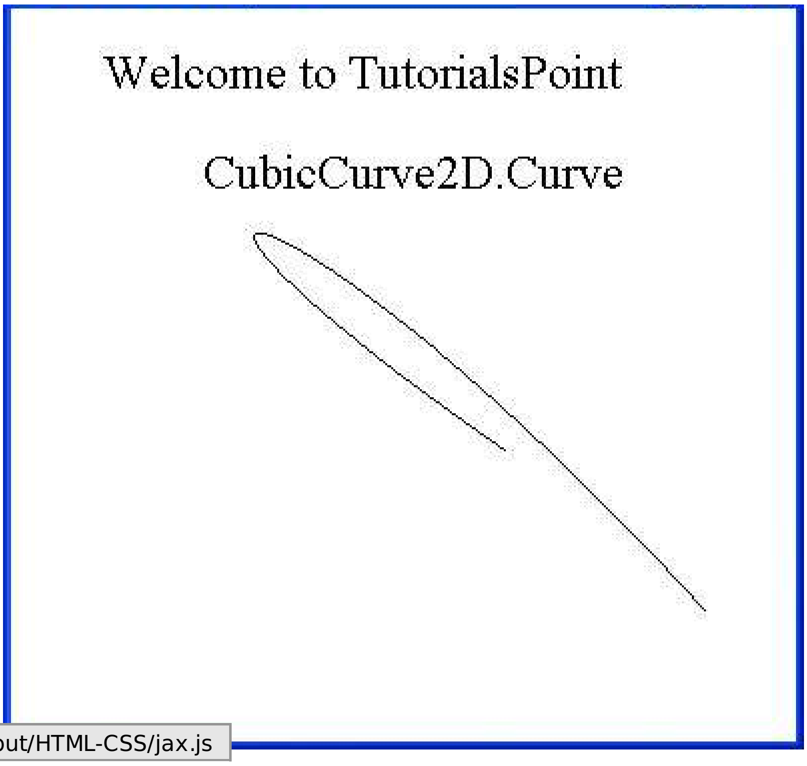
and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AWTGraphicsDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\AWT>java com.tutorialspoint.gui.AWTGraphicsDemo
```

Verify the following output



Welcome to TutorialsPoint

CubicCurve2D.Curve