JFREECHART - QUICK GUIDE

http://www.tutorialspoint.com/jfreechart/jfreechart_quick_guide.htm

A chart is a graphical representation of information. There are various tools available, which can be used to create different types of charts.

This tutorial helps you to understand what exactly the **JfreeChart** is, why is it required and the various ways to create different types of charts within a Java based application or independently.

What is JFreeChart?

JfreeChart is an open source library developed in Java, which can be used within Java based applications to create a wide range of charts. By using JFreeChart, we can create all the major type of 2D and 3D charts such as pie chart, bar chart, line chart, XY chart and 3D charts.

Why JFreeChart?

- It is open source and 100% free, which permits usage in the commercial applications without any cost.
- It comes with well documented APIs, which makes it quite easy to understand.
- It supports a wide range of chart types such as Pie Chart, Line Chart, Bar Chart, Area Chart and 3D charts.
- JFreeChart is easy to extend and can be used in both, the client-side, as well as the server-side applications.
- It supports multiple output formats like PNG, JPEG, PDF, SVG etc.
- It allows extensive customizations of charts.

Where do we use JFree Chart?

Consider a situation where you are developing an application and you need to show the data in the form of charts, where the data itself is populated dynamically. In such case, you can use JFreeChart to display the data in the form of charts using simple programming.

History

The JFreeChart project was founded fourteen years ago, in February 2000, by David Gilbert. Today, JFreeChart is the most widely used charting library among Java developers.

JFREECHART INSTALLATION

This chapter takes you through the process of setting up JFreeChart on Windows and Linux. User administration is needed while installing the JFreeChart. The JFreeChart is famous for its efficient chart creation, and user-friendly installation setup.

System Requirements :

JDK	1.5 or above
Memory	2GB RAM
Disk Space	No minimum requirement
Operating System Version	Linux or Windows

Installing JFreeChart

To install JFreeChart, first you need to install Java on your system.

Step1: Verifying Java Installation

To verify Java installation, open the console and execute the following **java** command :

Os	Task	Command
Windows	Open command console	C:>java -version
Linux	Open command terminal	$java -version

If Java installation is done properly, then you should get the following output for both the operating systems :

OS	Description
Windows	Java version "1.7.0_60" Java TM SE Run Time Environment $build1.7.0_{6}0 - b19$ Java HotSpotTM 64-bit Server VM $build24.60 - b09, mixedmode$
Linux	java version "1.7.0_25" OpenJDK Runtime Environment $rhel - 2.3.10.4.el6_{4} - x86_{6}4$ OpenJDK 64-Bit Server VM $build23.7 - b01, mixedmode$

If you do not have Java installed, then install the Java Software Development Kit SDK from the link:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

We assume that you have installed Java 1.7.0_60 version before proceeding for this tutorial.

Step 2: Setting JAVA Environment

Set the **JAVA_HOME** environment variable to point to the base directory location where Java is installed on your machine. For example,

Os	Description
Windows	Set Environmental varible JAVA_HOME to C:\ProgramFiles\java\jdk1.7.0_60
Linux	export JAVA_HOME=/usr/local/java-current

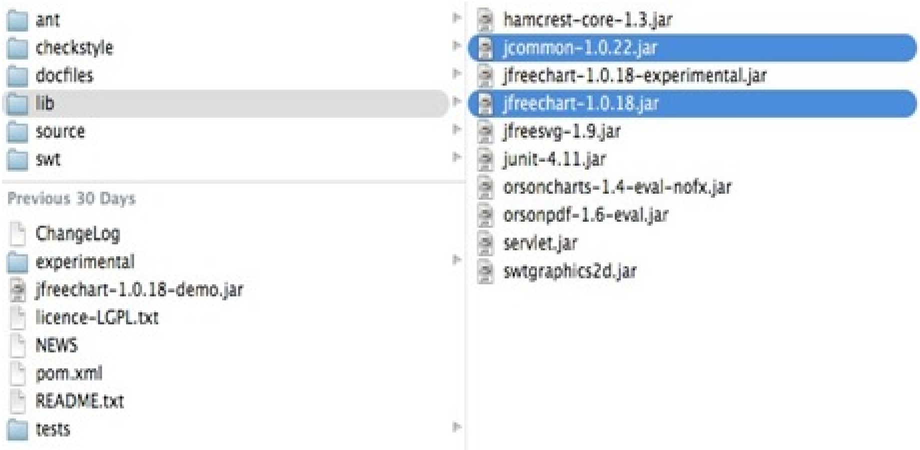
Append Java compiler location to System Path.

OS	Description
Windows	Append the String; C:\Program Files\Java\jdk1.7.0_60\bin to the end of the system variable PATH.
Linux	export PATH=$PATH$:JAVA_HOME/bin/

Verify the command **java -version** from command prompt as explained above

Step3: Installing JFreeChart

Download the latest version of JFreeChart.zip from the link http://www.jfree.org/jfreechart/download/ Unzip the downloaded file at any location from where required libraries can be linked into your Java program. The following image shows the structure of the directories and files :



ant
checkstyle
docfiles
lib
source
swt

Previous 30 Days

ChangeLog
experimental
jfreechart-1.0.18-demo.jar
licence-LGPL.txt
NEWS
pom.xml
README.txt
tests

hamcrest-core-1.3.jar
jcommon-1.0.22.jar
jfreechart-1.0.18-experimental.jar
jfreechart-1.0.18.jar
jfreesvg-1.9.jar
junit-4.11.jar
orsoncharts-1.4-eval-nofx.jar
orsonpdf-1.6-eval.jar
servlet.jar
swtgraphics2d.jar

Add complete path of jfreechart-1.0.18.jar and jcommon-1.0.22.jar files to the CLASSPATH as shown below :

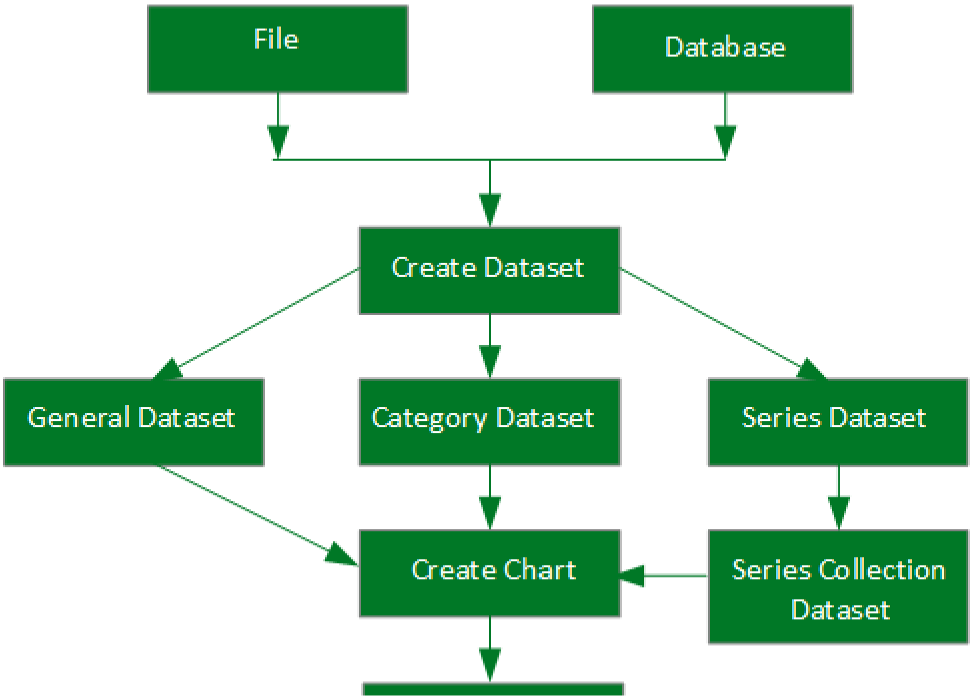
OS	Description
Windows	Append the Strings “C:\ jfreechart-1.0.18\lib\ jfreechart-1.0.18.jar” and “C:\ jfreechart-1.0.18\lib\ jcommon-1.0.22.jar” to the end of the user variable CLASSPATH
Linux	Export CLASSPATH=$CLASSPATH: /usr/share/jfreechart-1.0.18/lib/jfreechart-1.0.18.jar: /usr/share/jfreechart-1.0.18/lib/jcommon-1.0.22.jar

JFREECHART ARCHITECTURE

This chapter explains basic class level and application level architectures of JFreeChart to give you an idea about how JFreeChart interacts with different classes and how it fits in your Java based application.

Class Level Architecture

The class level architecture explains how various classes from the library interacts with each other to create various types of charts.

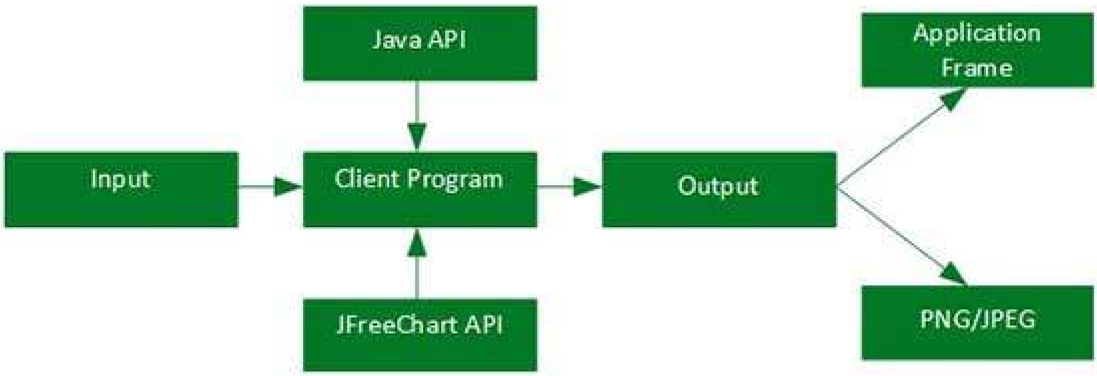


Following is the detail of the units used in the above block diagram :

Units	Description
File	The source having user input to be used for creating a dataset in the file.
Database	The source having user input to be used for creating a dataset in the database.
Create Dataset	Accepts the dataset and stores the dataset into dataset object.
General Dataset	This type of dataset is mainly used for pie charts.
Category Dataset	This type of dataset is used for bar chart, line chart,etc.
Series Dataset	This type of dataset is used for storing series of data and construct line charts.
Series Collection Dataset	The different categories of series datasets are added to series collection dataset. This type of dataset is used for XYLine Charts.
Create Chart	This is the method which is executed to create final chart.
Frame/Image	The chart is displayed on a Swing Frame or an image is created.

Application Level Architecture

The application level architecture explains where JFreeChart library sits inside a Java Application.



The client program receives user data and then it make use standard Java and JFreeChart APIs based on requirements to generate the output in the form of either a frame, which can be displayed directly inside the application or independently in the image formats such as JPEG or PNG.

JFREECHART - PIE CHART

In a pie chart, the arc length of each sector is proportional to the quantity it represents. This chapter demonstrates how we can use JFreeChart to create **Pie Chart** from a given set of business data.

Business data

The following example depicts mobile sale with the help of a pie chart. Following is a list of different mobile brands and their sale *unitsperday*.

S.N.	Mobile Brands	Sales *UNITSperday*
1	Iphone 5S	20
2	Samsung Grand	20
3	MOTO G	40
4	Nokia Lumia	10

AWT Based Application

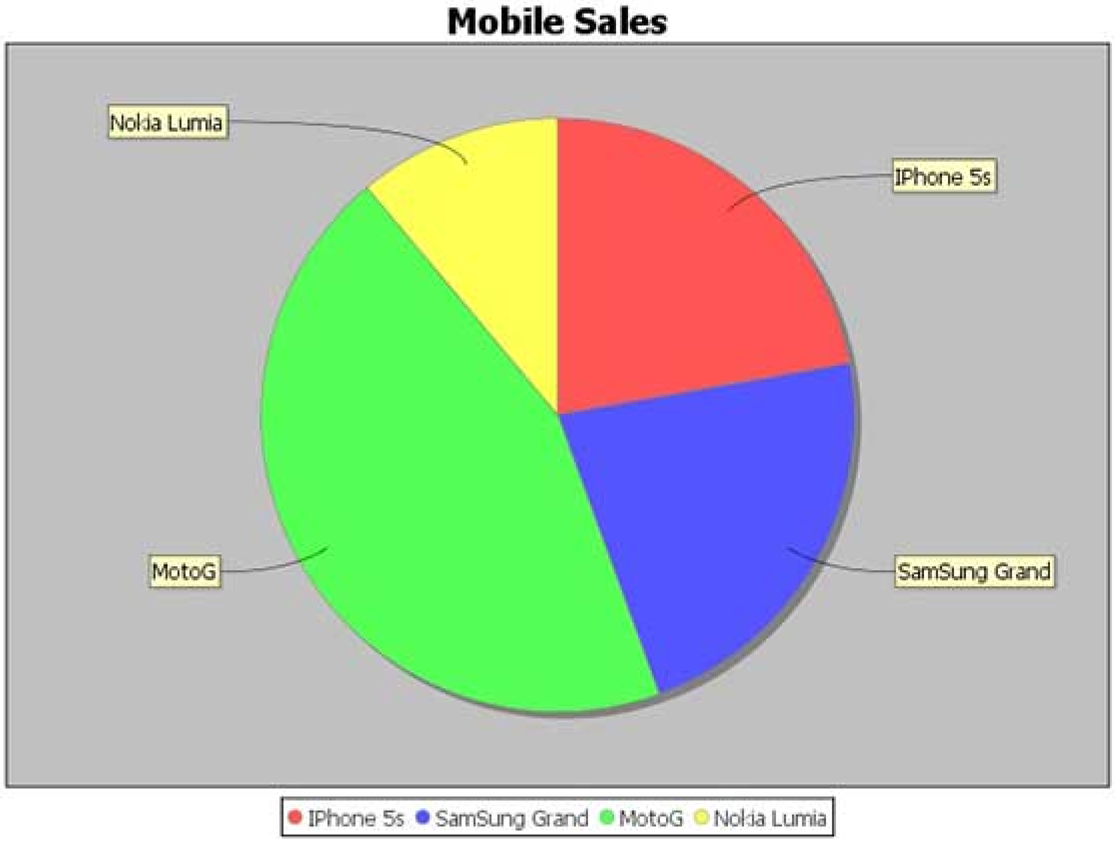
Following is the code to create Pie Chart from the above given information. This code helps you embed a pie chart in any AWT based application.

```java
import javax.swing.JPanel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;

public class PieChart_AWT extends ApplicationFrame
{
   public PieChart_AWT( String title )
   {
      super( title );
      setContentPane(createDemoPanel( ));
   }
   private static PieDataset createDataset( )
   {
      DefaultPieDataset dataset = new DefaultPieDataset( );
      dataset.setValue( "IPhone 5s" , new Double( 20 ) );
      dataset.setValue( "SamSung Grand" , new Double( 20 ) );
      dataset.setValue( "MotoG" , new Double( 40 ) );
      dataset.setValue( "Nokia Lumia" , new Double( 10 ) );
      return dataset;
   }
   private static JFreeChart createChart( PieDataset dataset )
   {
      JFreeChart chart = ChartFactory.createPieChart(
         "Mobile Sales",  // chart title
         dataset,        // data
         true,           // include legend
         true,
         false);

      return chart;
   }
   public static JPanel createDemoPanel( )
   {
      JFreeChart chart = createChart(createDataset( ) );
      return new ChartPanel( chart );
   }
   public static void main( String[ ] args )
   {
      PieChart_AWT demo = new PieChart_AWT( "Mobile Sales" );
      demo.setSize( 560 , 367 );
      RefineryUtilities.centerFrameOnScreen( demo );
      demo.setVisible( true );
   }
}
```

Let us keep above Java code in **PieChart_AWT.java** file, and then compile and run it from command prompt as follows :

```
$javac PieChart_AWT.java
$java PieChart_AWT
```

If everything is fine , it will compile and run to generate the following Pie Graph :



If you do not need to embed your chart in any application then you can create chart images at command prompt. JFreeChart allows you to save chart images in either JPG or PNG formats.

JPEG Image Creation

Let us re-write above example to generate a JPEG image from command line. Following are two APIs provided by JFreeChart library, which you can use to generate either PNG or JPEG image as per your requirement.

- **saveChartAsPNG -** API to save image in PNG format.
- **saveChartAsJPEG -** API to save image in JPEG format.

```
import java.io.*;
import org.jfree.chart.ChartUtilities;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.data.general.DefaultPieDataset;

public class PieChart {
   public static void main( String[ ] args ) throws Exception
   {

      DefaultPieDataset dataset = new DefaultPieDataset( );
      dataset.setValue("IPhone 5s", new Double( 20 ) );
      dataset.setValue("SamSung Grand", new Double( 20 ) );
      dataset.setValue("MotoG", new Double( 40 ) );
      dataset.setValue("Nokia Lumia", new Double( 10 ) );

      JFreeChart chart = ChartFactory.createPieChart(
         "Mobile Sales", // chart title
         dataset, // data
         true, // include legend
         true,
         false);

      int width = 640; /* Width of the image */
      int height = 480; /* Height of the image */
      File pieChart = new File( "PieChart.jpeg" );
      ChartUtilities.saveChartAsJPEG( pieChart , chart , width , height );
   }
}
```

Let us keep the above Java code in **PieChart.java** file, and then compile and run it from command prompt as follows :

```
$javac PieChart.java
$java PieChart
```

If everything is fine , it will compile and run to create JPEG image file named **PieChart.jpeg** in your current directory.

JFREECHART - BAR CHART

This chapter demonstrates how we can use JFreeChart to create **Bar Chart** from a given set of business data.

A bar chart uses different orientation *horizontalorvertical* bars to show comparisons in various categories. One axis *domainaxis* of the chart shows the specific domain being compared, and the other axis *rangeaxis* represents discrete values.

Business Data

The following example depicts various car statistics with the help of a bar chart. Following is a list of car brands along with their different characteristics, which we will show using a bar chart :

Car	Speed	User Rating	Millage	Safety
Fiat	1.0	3.0	5.0	5.o
Audi	5.0	6.0	10.0	4.0
Ford	4.0	2.0	3.0	6.0

AWT Based Application

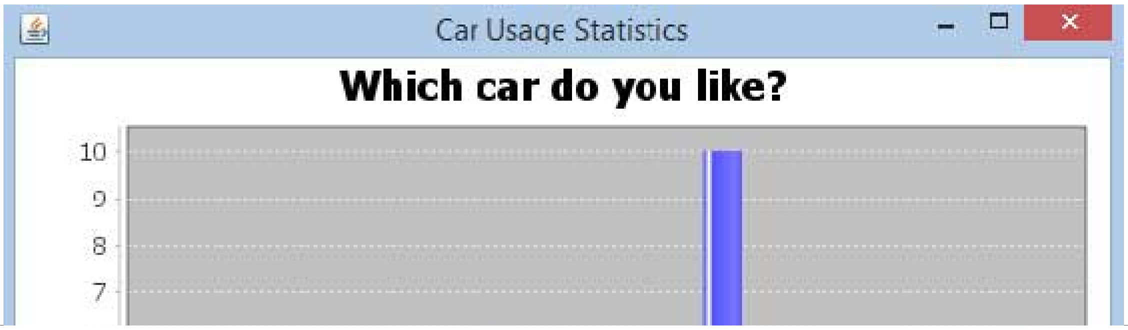
Following is the code to create Bar Chart from the above given information. This code helps you embed a bar chart in any AWT based application.

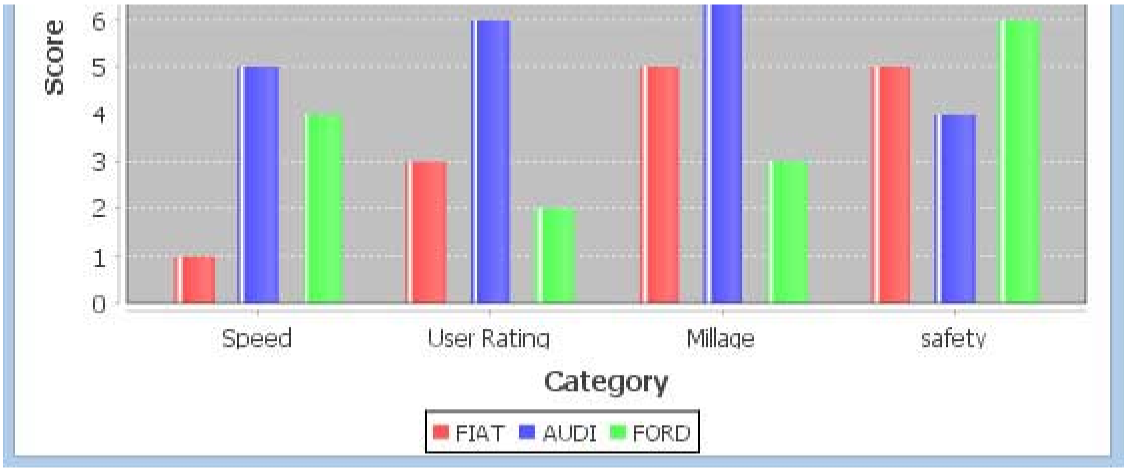
```
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.ui.ApplicationFrame;
import org.jfree.ui.RefineryUtilities;

public class BarChart_AWT extends ApplicationFrame
{
   public BarChart_AWT( String applicationTitle , String chartTitle )
   {
      super( applicationTitle );
      JFreeChart barChart = ChartFactory.createBarChart(
          chartTitle,
          "Category",
          "Score",
          createDataset(),
          PlotOrientation.VERTICAL,
          true, true, false);

      ChartPanel chartPanel = new ChartPanel( barChart );
      chartPanel.setPreferredSize(new java.awt.Dimension( 560 , 367 ) );
      setContentPane( chartPanel );
   }
   private CategoryDataset createDataset( )
   {
      final String fiat = "FIAT";
      final String audi = "AUDI";
      final String ford = "FORD";
      final String speed = "Speed";
      final String millage = "Millage";
      final String userrating = "User Rating";
      final String safety = "safety";
      final DefaultCategoryDataset dataset =
      new DefaultCategoryDataset( );

      dataset.addValue( 1.0 , fiat , speed );
      dataset.addValue( 3.0 , fiat , userrating );
      dataset.addValue( 5.0 , fiat , millage );
      dataset.addValue( 5.0 , fiat , safety );

      dataset.addValue( 5.0 , audi , speed );
      dataset.addValue( 6.0 , audi , userrating );
      dataset.addValue( 10.0 , audi , millage );
      dataset.addValue( 4.0 , audi , safety );

      dataset.addValue( 4.0 , ford , speed );
      dataset.addValue( 2.0 , ford , userrating );
      dataset.addValue( 3.0 , ford , millage );
      dataset.addValue( 6.0 , ford , safety );

      return dataset;
   }
   public static void main( String[ ] args )
   {
      BarChart_AWT chart = new BarChart_AWT("Car Usage Statistics", "Which car do you
like?");
      chart.pack( );
      RefineryUtilities.centerFrameOnScreen( chart );
      chart.setVisible( true );
   }
}
```

Let us keep the above Java code in **BarChart_AWT.java** file, and then compile and run it from command prompt as follows :

```
$javac BarChar_AWT.java
$java BarChart_AWT
```

If everything is fine , it will compile and run to generate the following Bar Graph :





JPEG Image Creation

Let us re-write above example to generate a JPEG image from command line.

```
import java.io.*;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.chart.ChartUtilities;

public class BarChart
{
   public static void main( String[ ] args )throws Exception
   {

      final String fiat = "FIAT";
      final String audi = "AUDI";
      final String ford = "FORD";
      final String speed = "Speed";
      final String millage = "Millage";
      final String userrating = "User Rating";
      final String safety = "safety";

      final DefaultCategoryDataset dataset = new DefaultCategoryDataset( );

      dataset.addValue( 1.0 , fiat , speed );
      dataset.addValue( 3.0 , fiat , userrating );
      dataset.addValue( 5.0 , fiat , millage );
      dataset.addValue( 5.0 , fiat , safety );

      dataset.addValue( 5.0 , audi , speed );
      dataset.addValue( 6.0 , audi , userrating );
      dataset.addValue( 10.0 , audi , millage );
      dataset.addValue( 4.0 , audi , safety );

      dataset.addValue( 4.0 , ford , speed );
      dataset.addValue( 2.0 , ford , userrating );
      dataset.addValue( 3.0 , ford , millage );
      dataset.addValue( 6.0 , ford , safety );

      JFreeChart barChart = ChartFactory.createBarChart(
         "CAR USAGE STATIStICS",
         "Category", "Score",
         dataset,PlotOrientation.VERTICAL,
         true, true, false);

      int width = 640; /* Width of the image */
      int height = 480; /* Height of the image */
      File BarChart = new File( "BarChart.jpeg" );
      ChartUtilities.saveChartAsJPEG( BarChart , barChart , width , height );
   }
}
```

Let us keep the above Java code in **BarChart.java** file, and then compile and run it from command prompt as follows:

```
$javac BarChart.java
$java BarChart
```

If everything is fine , it will compile and run to create JPEG image file named **BarChart.jpeg** in your current directory.