**GUI18 ASCII Chart**

**Directions**

The String class provides a method named **format** that will allow you to format output the same way as the **printf.**The benefit of this method is that we can use it to format the text we want to display within a textarea. Look at the example of the printf statement below.

int quantity = 3;

String descrip = "Hamburger";

double price = 5.25;

System.out.printf("%4d%12s%8f%n", quantity, descrip, price);

This code would produce the following output on the console window:

3 Hamburger 5.25

To display this same text within a textarea window you would do the following:

int quantity = 3;

String descrip = "Hamburger";

double price = 5.25;

String text =

String.format

("%4d%12s%8f%n", quantity, descrip, price);

textArea.setText(text);

Copy the following code to your source file.

import java.util.\*;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

public class GUI18 extends JFrame implements ActionListener

{

// instance variables

private JButton button;

private JTextArea textArea;

private JScrollPane scrollpane;

// constructor

public GUI18()

{

// create button

button = new JButton("run");

// set button attributes

button.setLocation(220,20);

button.setSize(80, 25);

// add button to frame

getContentPane().add(button);

// register listener with button

button.addActionListener(this);

// create textarea

textArea = new JTextArea();

// set textarea attributes

textArea.setFont(new Font(

"consolas"

, Font.PLAIN, 16));

textArea.setEditable(false); // cannot type text into textArea

textArea.setLineWrap(true); // wrap text when reach right border

// create scrollpane and add textarea

scrollpane = new JScrollPane(textArea);

// set scrollpane attributes

scrollpane.setLocation(50,50);

scrollpane.setSize(400, 400);

scrollpane.setBorder(BorderFactory.createLineBorder (Color.blue, 2));

// add scrollpane to frame

getContentPane().add(scrollpane);

// set frame attributes

setLayout(null);

setSize(500, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

public void actionPerformed(ActionEvent event)

{

}

// main method

public static void main(String[] args)

{

GUI18 app = new GUI18(); // run program

}

}

**Modifications**

Modify the **actionPerformed** method so that the program displays an ASCII chart in three columns like the Sample Run below.

1. The chart consists of series of ASCII values (integers) followed by their corresponding character displayed in three columns.
2. The ASCII values 32 - 63 should be in the first column, ASCII values 64 - 95 in the second column, and ASCII values 96 - 127 in the third column.
3. You will need to use the String class's format method to display the chart in three columns.
4. Recall that to convert an ASCII integer value into its character representation you have to use casting:

int num = 32;

char ch = (char)num; // convert int to char

1. The format specifier **%s** can be used with character data.

**Source File**

GUI18.java

**Sample Run**

