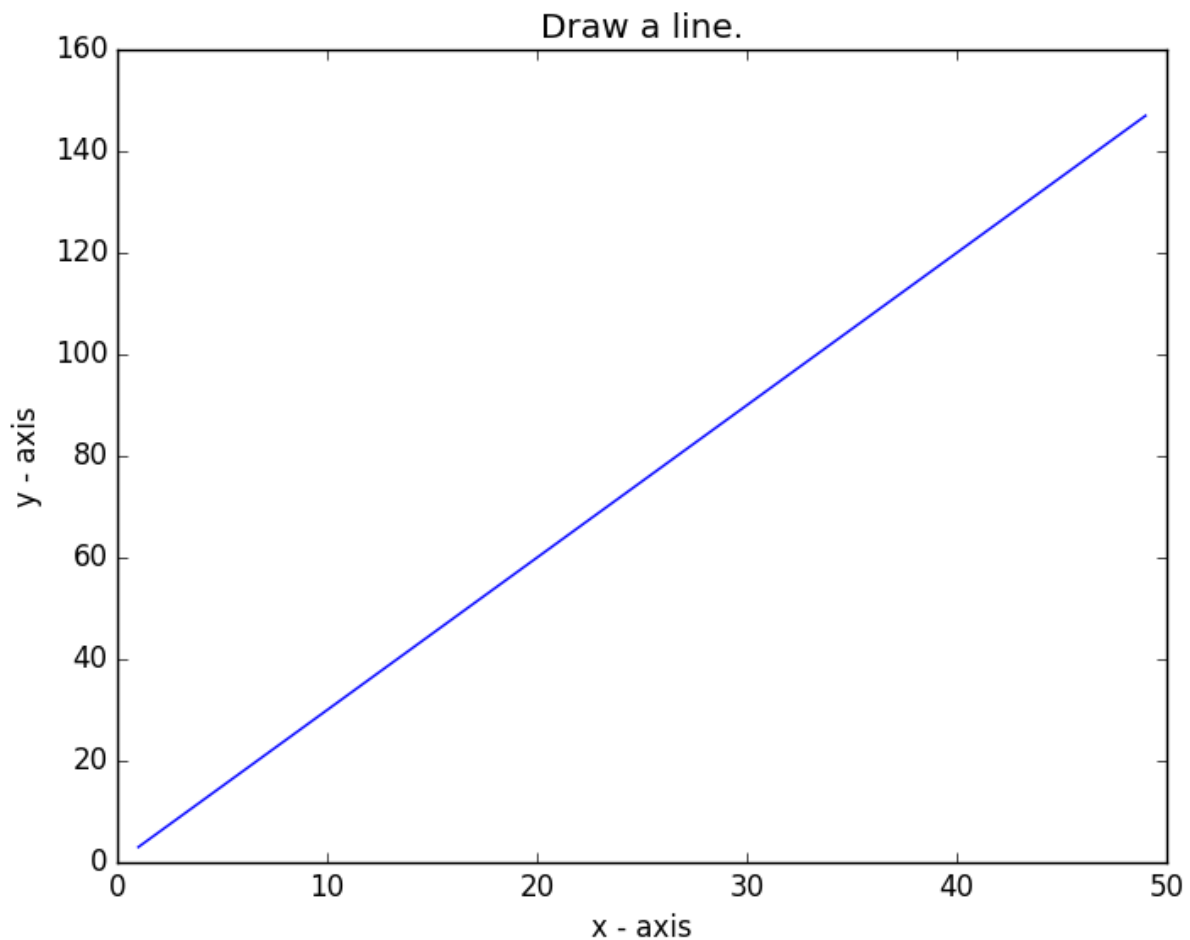


Assignment 4: Matplotlib Exercises

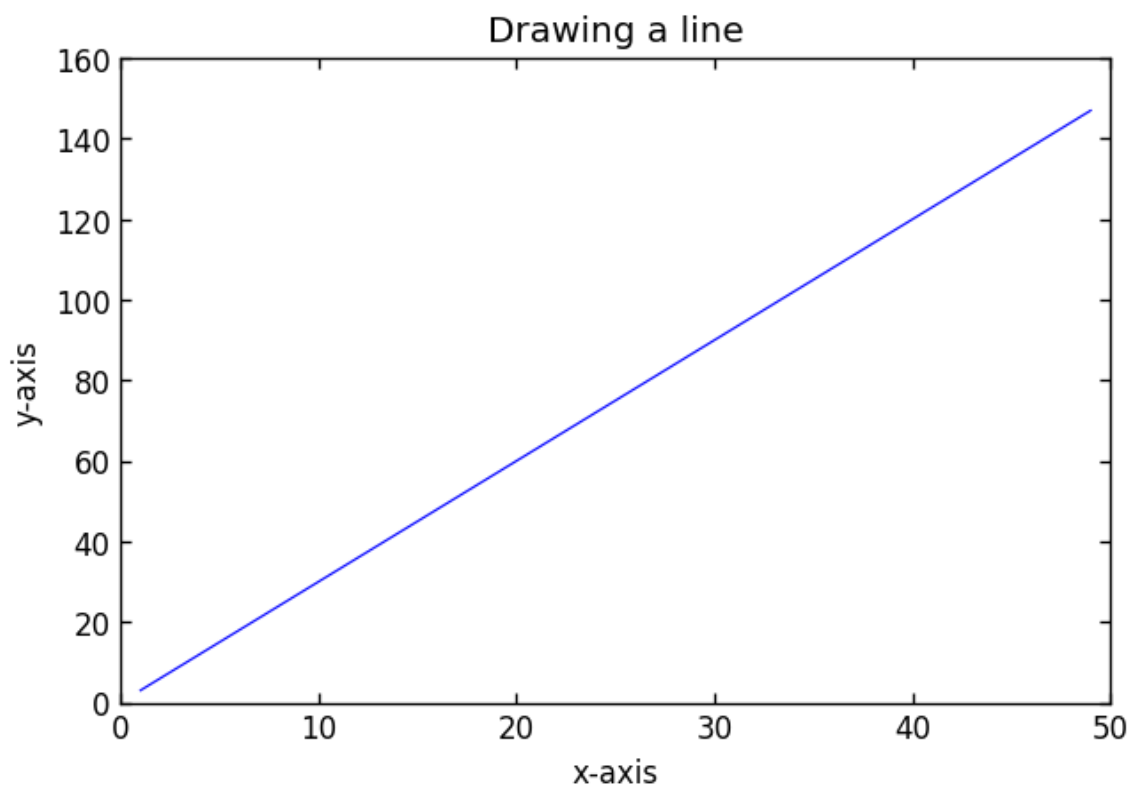
by Arnav Kumar Jain

1. Draw a line as shown below (value in y axis are thrice the value in x-axis). Add a suitable label in the x axis, y axis and a title.

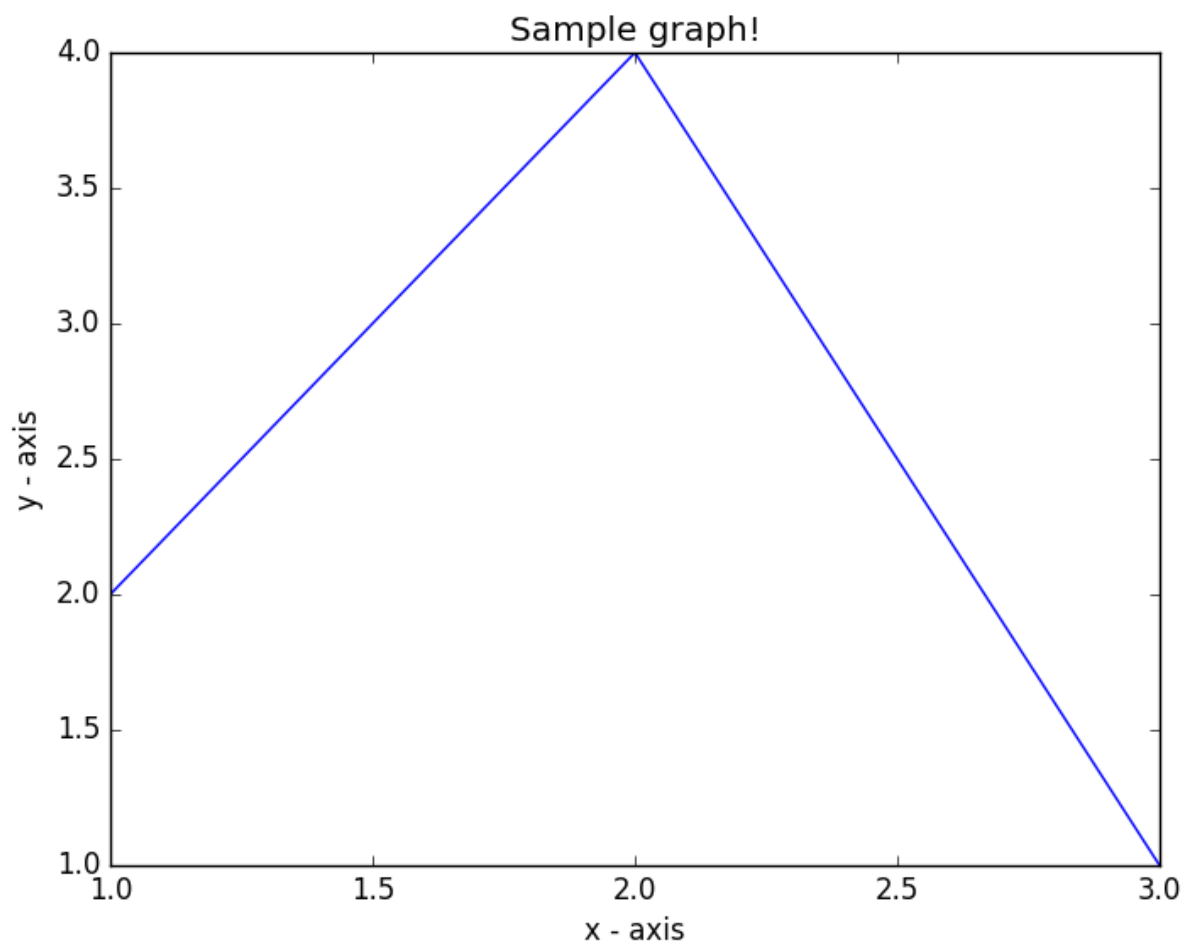


In [95]:

```
import matplotlib.pyplot as plt
def graph():
    x=[i for i in range(1,50)]
    y=[3*i for i in x]
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True)
    plt.plot(x,y,color="blue",lw=0.7)
    plt.xlabel("x-axis")
    plt.ylabel("y-axis")
    plt.title('Drawing a line')
    plt.xticks([i for i in range (0,51,10)])
    plt.yticks([i for i in range(0,161,20)])
    plt.axis([0,50,0,160])
    plt.show()
if __name__=="__main__":
    graph()
```

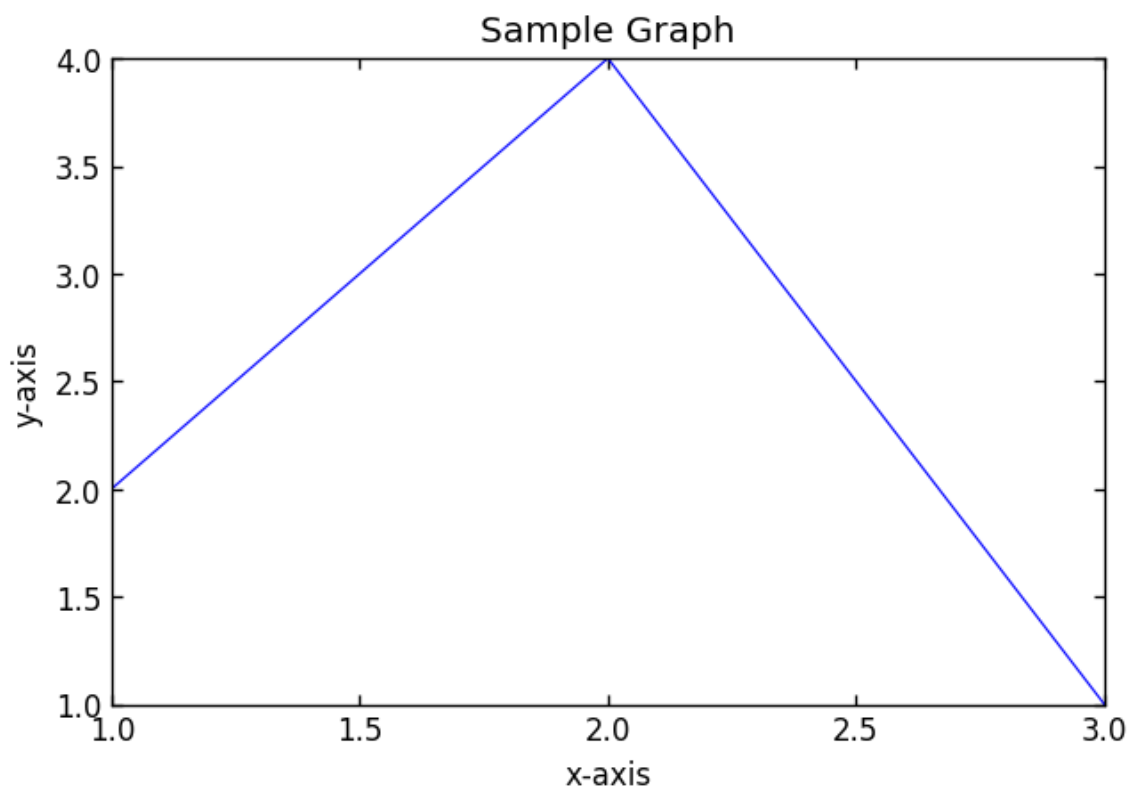


2. Write a Python program to draw following output



In [96]:

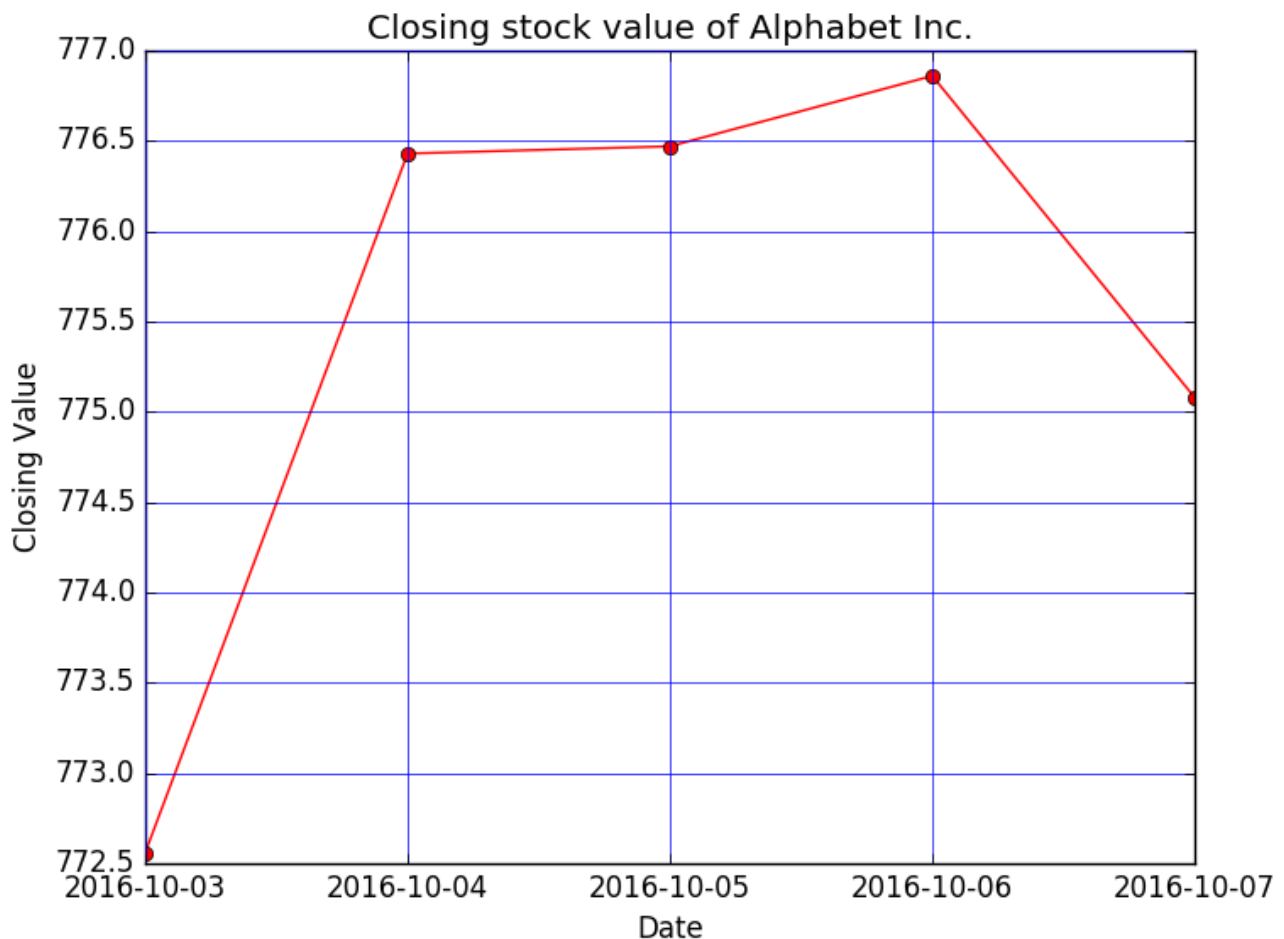
```
import matplotlib.pyplot as plt
def graph():
    x=[1,2,3]
    y=[2,4,1]
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True)
    plt.plot(x,y,'-',color="blue",lw=0.7)
    plt.axis([1,3,1,4])
    plt.xlabel("x-axis")
    plt.ylabel("y-axis")
    plt.title('Sample Graph')
    plt.xticks([0.5*i for i in range(2,7)])
    plt.show()
if __name__=="__main__":
    graph()
```



3. Write a Python program to display the grid and draw line charts of the closing value of ABC Ltd. between October 3, 2019 to October 7, 2019. Customized the grid lines with linestyle -, width .5. and color blue.

Date	Close
03-10-16	772.559998
04-10-16	776.429993
05-10-16	776.469971
06-10-16	776.859985
07-10-16	775.080017

The code snippet gives the output shown in the following screenshot:

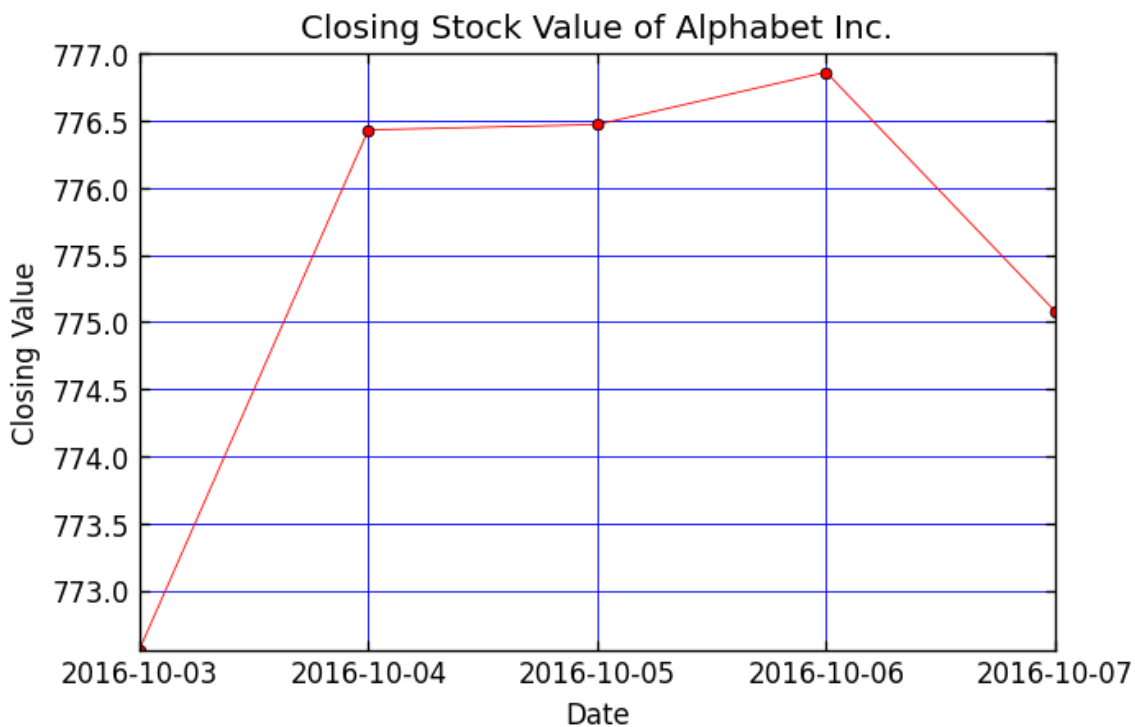


In [97]:

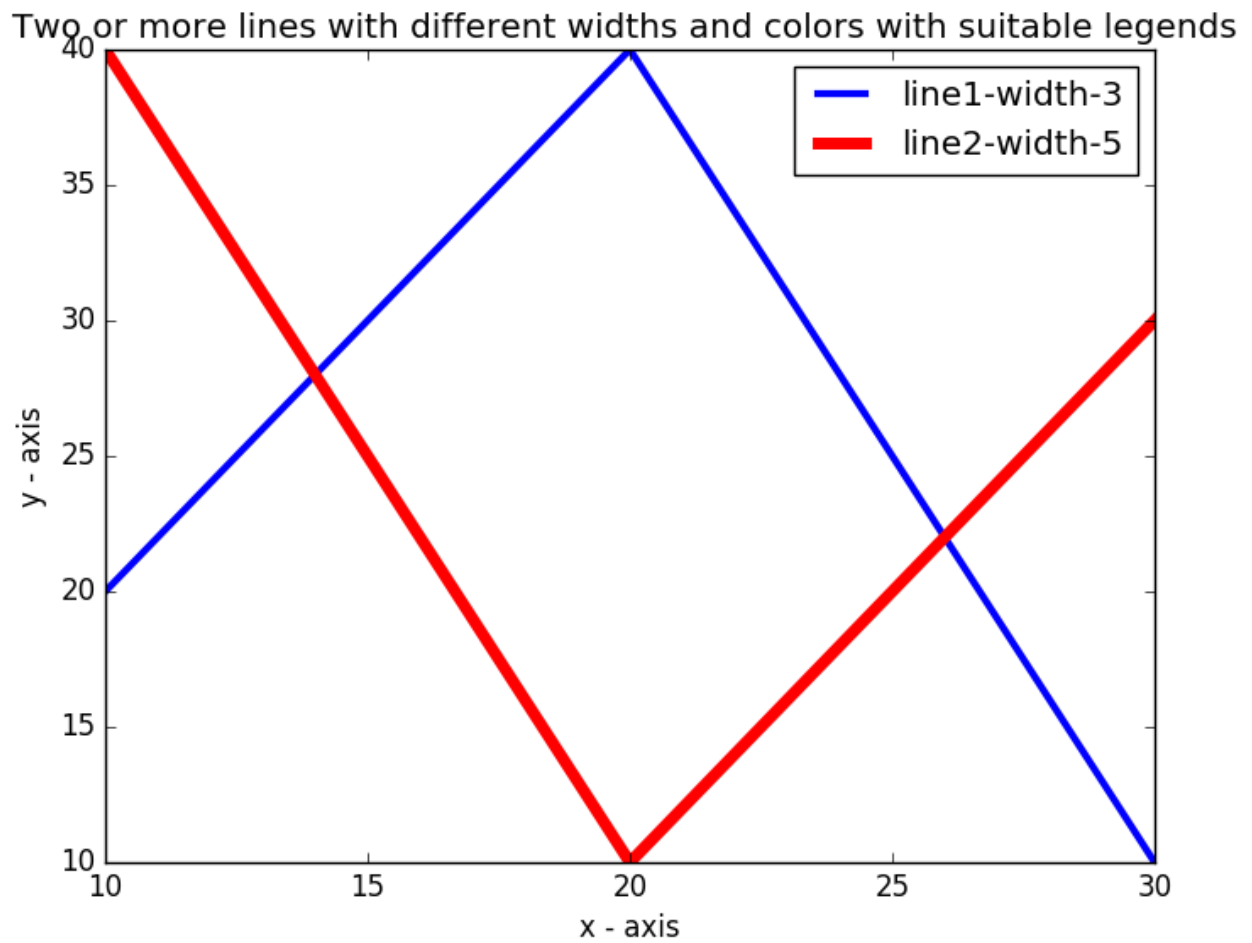
```

import matplotlib.pyplot as plt
import datetime as dt
def graph():
    x=['03-10-16','04-10-16','05-10-16','06-10-16','07-10-16']
    x=[dt.datetime.strptime(i,'%d-%m-%y') for i in x]
    x=[i.strftime('%Y-%m-%d') for i in x]
    y=[772.559998,776.429993,776.469971,776.859985,775.080017]
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True)
    plt.grid(color='b',ls='--',lw=0.5)
    plt.plot(x,y,'o-',color="red",lw=0.5,mec='k',ms=4,mew=0.5)
    plt.xlabel("Date")
    plt.ylabel("Closing Value")
    plt.title('Closing Stock Value of Alphabet Inc.')
    plt.axis([x[0],x[-1],y[0],777.0])
    plt.show()
if __name__=="__main__":
    graph()

```



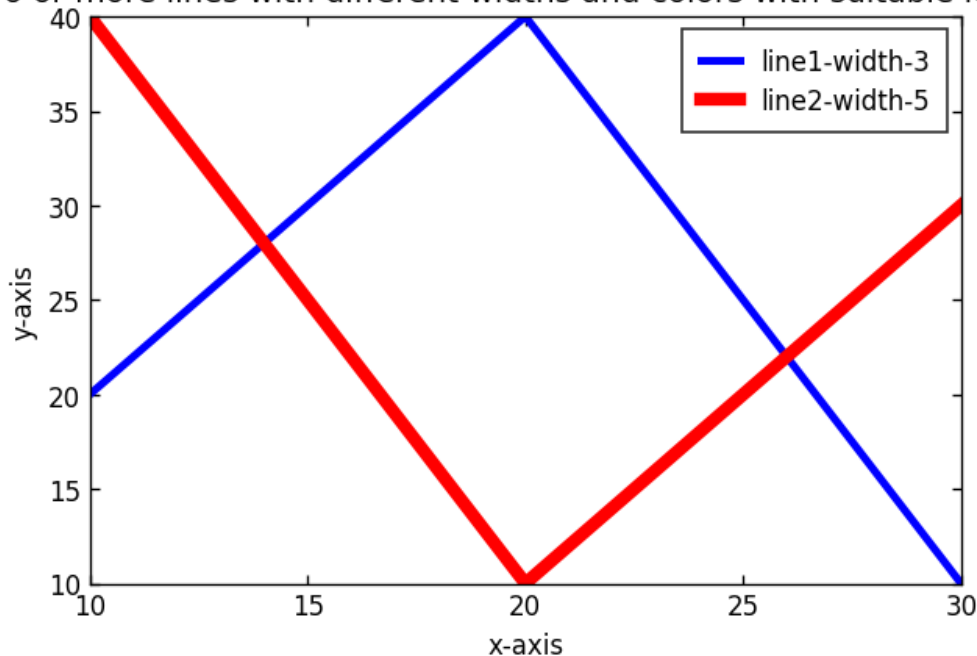
4. Plot two lines as shown below with appropriate legends, different widths and colors.



In [98]:

```
import matplotlib.pyplot as plt
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True)
    plt.plot([10,20,30],[20,40,10], '--',color="blue",lw=3,label="line1-width-3")
    plt.plot([10,20,30],[40,10,30], '--',color="red",lw=5,label='line2-width-5')
    plt.axis([10,30,10,40])
    plt.xlabel("x-axis")
    plt.ylabel("y-axis")
    plt.xticks([i for i in range(10,31,5)])
    plt.title('Two or more lines with different widths and colors with suitable legends')
    plt.legend(edgecolor='black',fancybox=False,borderpad=0.7,handletextpad=0.8,handlenght=1.5,labelsapcing=0.4)
    plt.show()
if __name__=="__main__":
    graph()
```

Two or more lines with different widths and colors with suitable legends



5. Display a bar chart of the popularity of programming Languages. Attach a text label above each bar displaying its popularity (float value). Use different color for each bar. Make blue border to each bar.

Sample data:

Programming languages: Java, Python, PHP, JavaScript, C#, C++

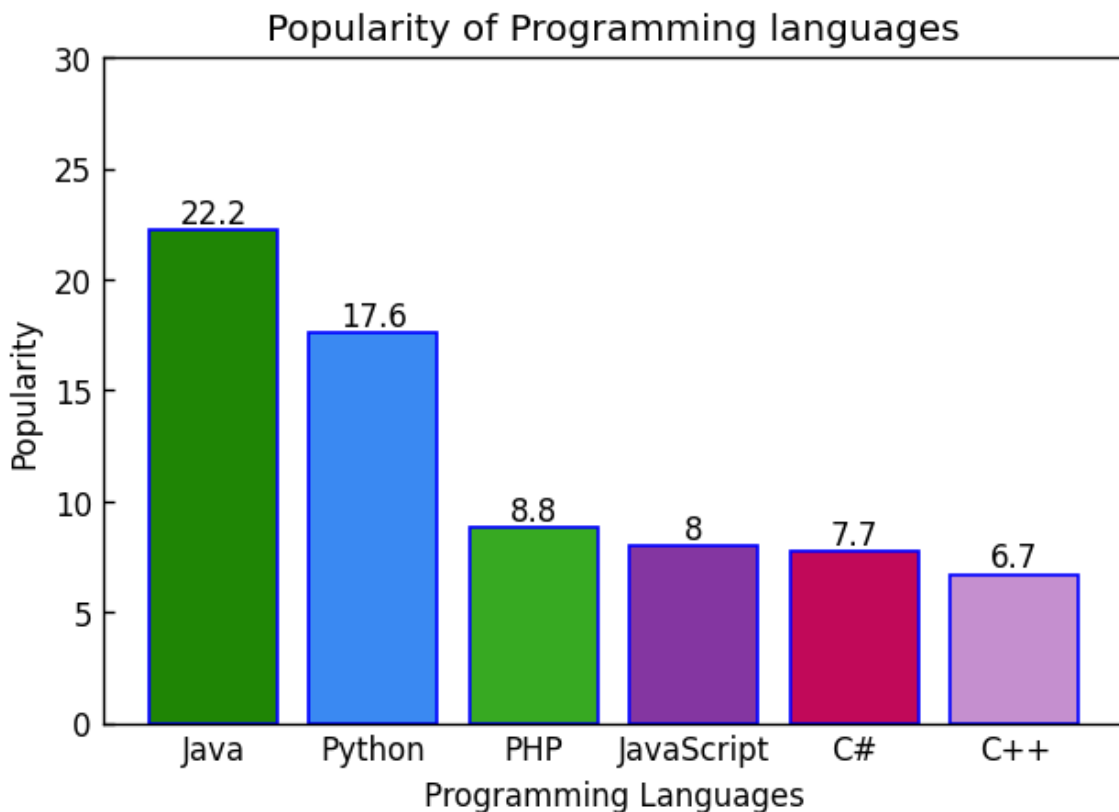
Popularity: 22.2, 17.6, 8.8, 8, 7.7, 6.7

In [114]:

```

import matplotlib.pyplot as plt
import random
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=False,right=False,bottom=False)
    x=['Java','Python','PHP','JavaScript','C#','C++']
    y=[22.2, 17.6, 8.8, 8, 7.7, 6.7]
    color=[(random.random(),random.random(),random.random()) for i in x]
    plt.bar(x,y,color=color,ec='b')
    for i in range(len(x)):
        plt.annotate(y[i],xy=(x[i],y[i]),ha="center",va="bottom")
    plt.title("Popularity of Programming languages")
    plt.xlabel("Programming Languages")
    plt.ylabel("Popularity")
    plt.ylim(top=30)
    plt.show()
if __name__=="__main__":
    graph()

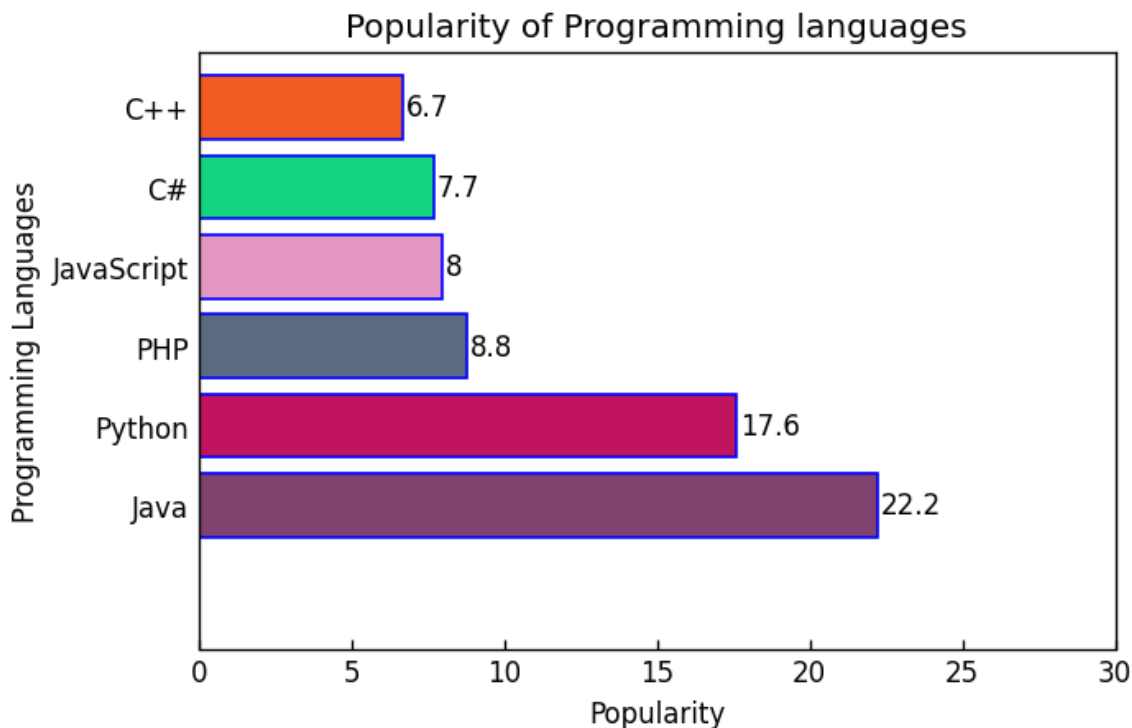
```



6. Convert bar chart of Q5 into horizontal bar chart and increase its bottom margin.

In [113]:

```
import matplotlib.pyplot as plt
import random
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=False,right=False,left=False)
    x=['Java','Python','PHP','JavaScript','C#','C++']
    y=[22.2, 17.6, 8.8, 8, 7.7, 6.7]
    color=[(random.random(),random.random(),random.random()) for i in x]
    plt.barh(x,y,color=color,ec='b')
    for i in range(len(x)):
        plt.text(s=y[i],x=y[i]+0.1,y=i,va='center',ha="left")
    plt.title("Popularity of Programming languages")
    plt.ylabel("Programming Languages")
    plt.xlabel("Popularity")
    plt.xlim(right=30)
    plt.ylim(bottom=-1.8)
    plt.show()
if __name__=="__main__":
    graph()
```

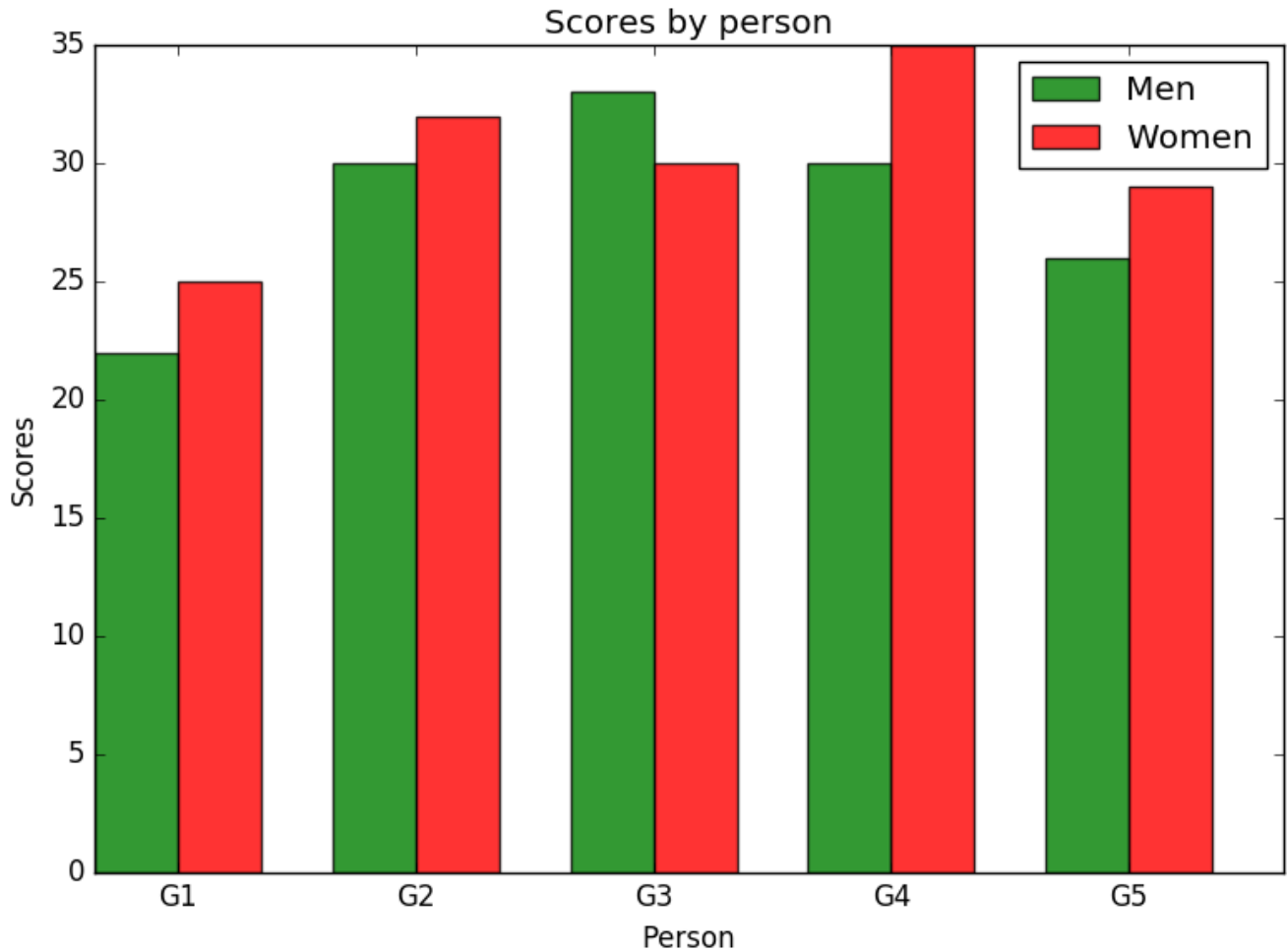


7. Create bar plot of scores by group and gender. Use multiple X values on the same chart for men and women.

Means (men) = (22, 30, 35, 35, 26)

Means (women) = (25, 32, 30, 35, 29)

Output should be as follows:



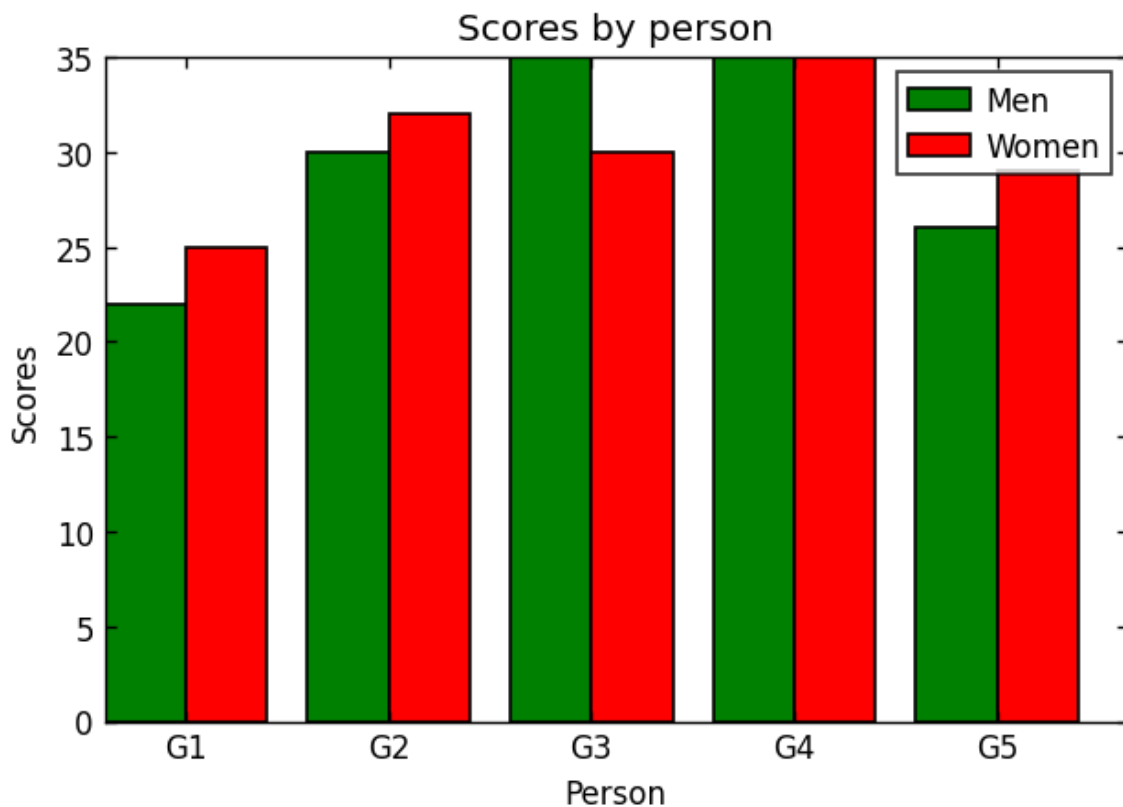
In [101]:

```

import matplotlib.pyplot as plt
import random
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True,bottom=True)
    x=['G1','G2','G3','G4','G5']
    Men=(22, 30, 35, 35, 26)
    Women=(25, 32, 30, 35, 29)
    b1=plt.bar([i-0.20 for i in range(len(x))],Men,color='g',ec='k',width=0.4)
    b2=plt.bar([i+0.20 for i in range(len(x))],Women,color='r',ec='k',width=0.4)
    plt.title("Scores by person")
    plt.xlabel("Person")
    plt.ylabel("Scores")
    plt.xlim(left=0-0.40)
    plt.xticks(range(len(x)),labels=x)
    plt.legend([b1,b2],['Men','Women'],edgecolor='black',fancybox=False,borderpad=0.4,handletextpad=0.5,labels spacing=0.4)

    plt.ylim(top=35)
    plt.show()
if __name__=="__main__":
    graph()

```



8. Write a Python program to create stack bar plot and add label to each section.

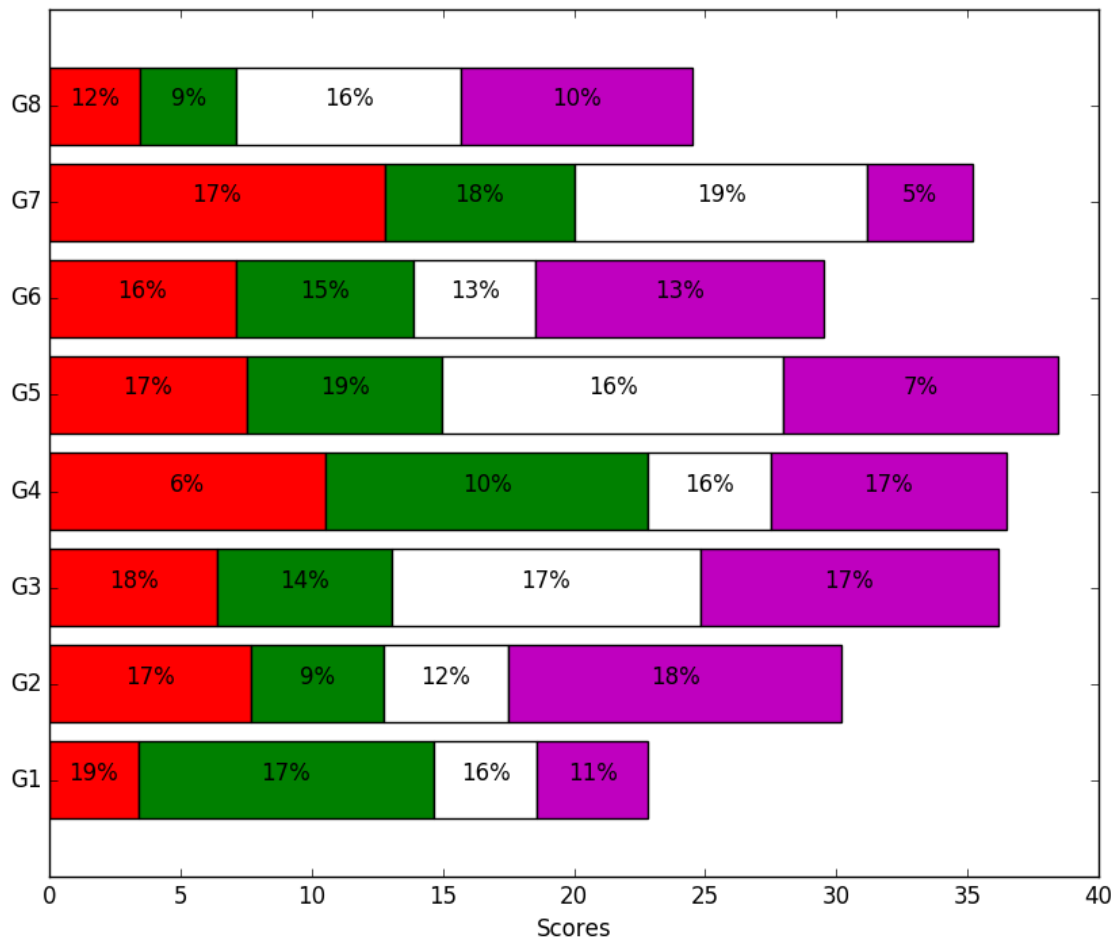
Sample data:

```
people = ('G1','G2','G3','G4','G5','G6','G7','G8')
```

```
segments = 4
```

Multi-dimensional data

```
data = [[ 3.40022085, 7.70632498, 6.4097905, 10.51648577, 7.5330039, 7.1123587, 12.77792868,
3.44773477], [ 11.24811149, 5.03778215, 6.65808464, 12.32220677, 7.45964195, 6.79685302,
7.24578743, 3.69371847], [ 3.94253354, 4.74763549, 11.73529246, 4.6465543, 12.9952182,
4.63832778, 11.16849999, 8.56883433], [ 4.24409799, 12.71746612, 11.3772169, 9.00514257,
10.47084185, 10.97567589, 3.98287652, 8.80552122]]
```

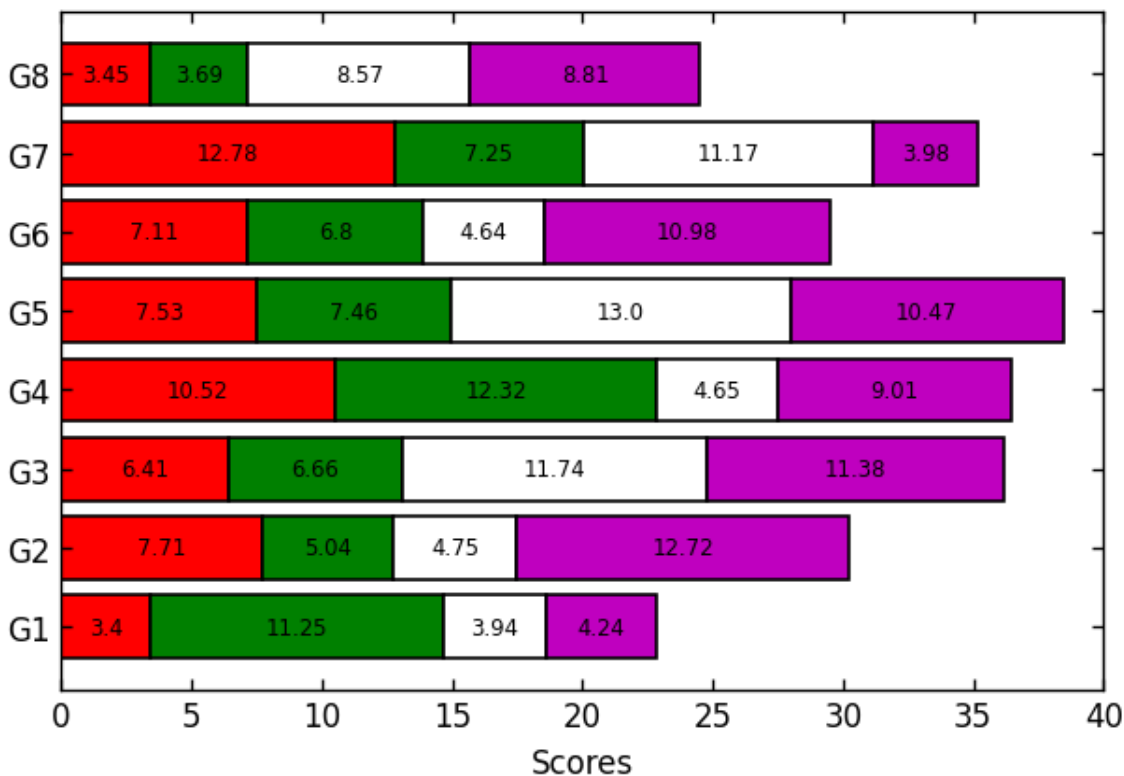


In [102]:

```

import matplotlib.pyplot as plt
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True,left=True)
    people = ('G1','G2','G3','G4','G5','G6','G7','G8')
    segments = 4
    color="rgwm"
    data = [[3.40022085, 7.70632498, 6.4097905, 10.51648577, 7.5330039, 7.1123587, 1
2.77792868, 3.44773477], [ 11.24811149, 5.03778215, 6.65808464, 12.32220677, 7.45964
195, 6.79685302, 7.24578743, 3.69371847], [ 3.94253354, 4.74763549, 11.73529246, 4.6
465543, 12.9952182, 4.63832778, 11.16849999, 8.56883433], [ 4.24409799, 12.71746612,
11.3772169, 9.00514257, 10.47084185, 10.97567589, 3.98287652, 8.80552122]]
    for i in range(len(data)):
        plt.barh(people,data[i],color=color[i],ec='k',height=0.8,left=0 if i==0 else
[sum(x for x in zip(*data[:i]))]
        for j in range(len(data[i])):
            plt.annotate(s=round(data[i][j],2),xy=((0 if i==0 else [sum(x for x in
zip(*data[:i]))][j])+data[i][j]/2,j),va='center',ha='center',size=7)
    plt.xlabel("Scores")
    plt.xlim(left=0,right=40)
    plt.show()
if __name__=="__main__":
    graph()
    print()

```

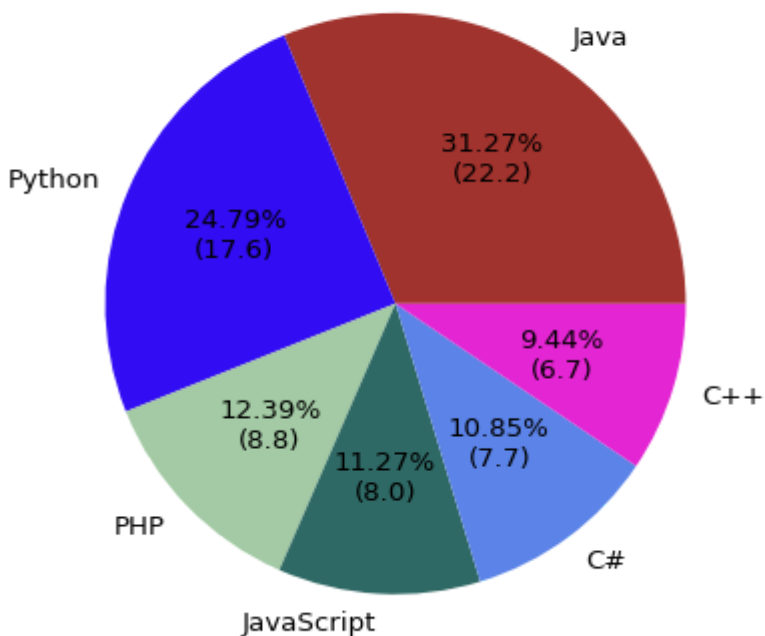


9. Create a pie chart of the popularity of programming Languages. Add a title to this pie chart. Use sample data given in Q5.

In [111]:

```
import matplotlib.pyplot as plt
import random
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=False,right=False,bottom=False)
    x=['Java','Python','PHP','JavaScript','C#','C++']
    y=[22.2, 17.6, 8.8, 8, 7.7, 6.7]
    color=[(random.random(),random.random(),random.random()) for i in x]
    plt.pie(y,labels=x,colors=color,autopct=lambda p: '{:.2f}%\n({:1.1f})'.format(p,
(p/100)*sum(y)),textprops={'size':8})
    plt.title("Popularity of Programming languages")
    plt.show()
if __name__=="__main__":
    graph()
    print()
```

Popularity of Programming languages



10. Create a pie chart of gold medal achievements of five most successful countries in 2016 Summer Olympics. Read the data from a csv file.

Sample data: medal.csv

country	gold_medal
United States	46
Great Britain	27
China	26
Russia	19
Germany	17

In [104]:

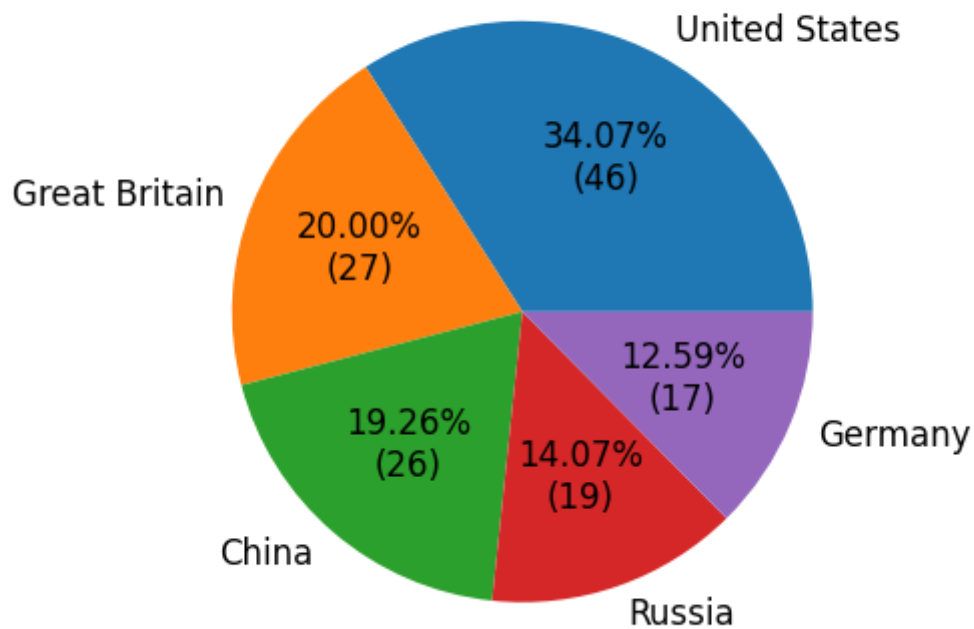
```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

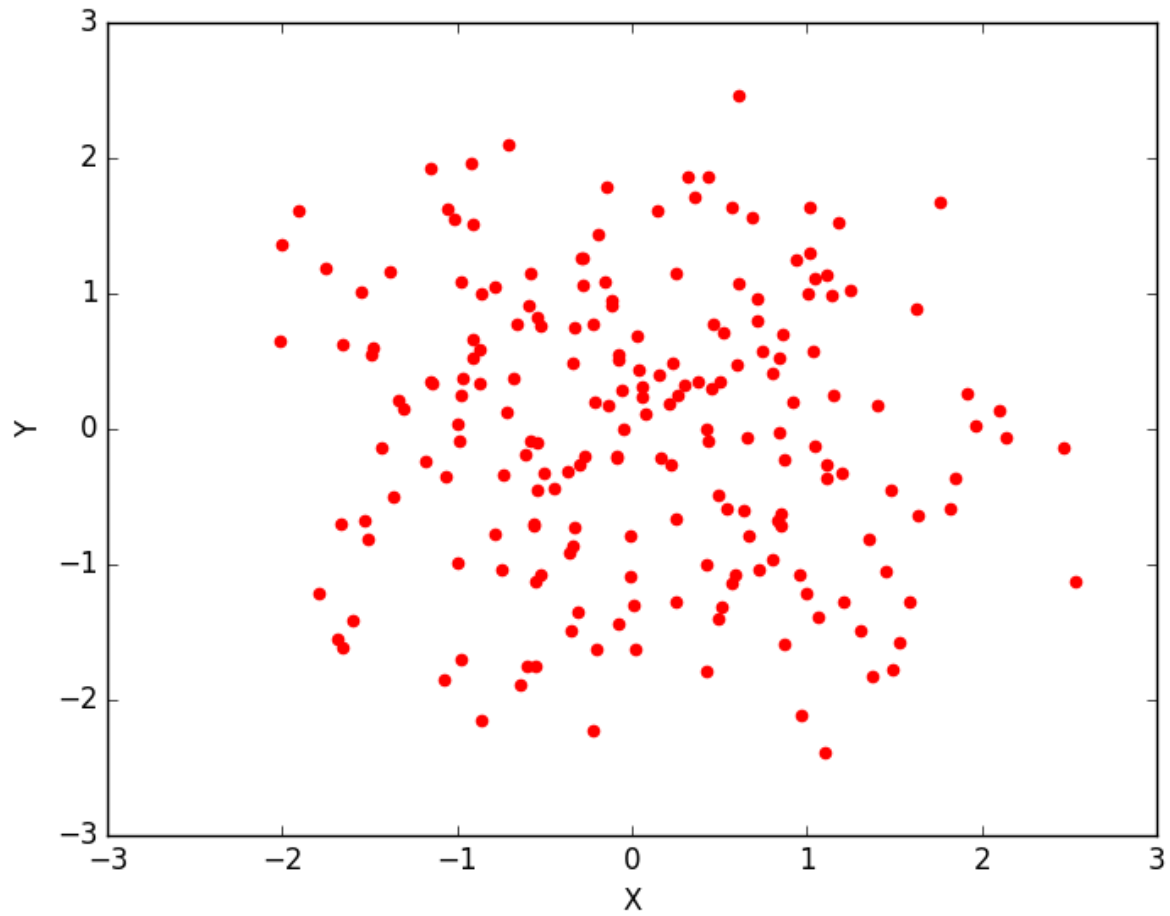
In [105]:

```
import matplotlib.pyplot as plt
import pandas as pd
def graph():
    path="/content/drive/MyDrive/Colab Notebooks/medal.csv"
    df=pd.read_csv(path)
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=False,right=False,bottom=False)
    plt.pie(df.iloc[:,1],labels=df.iloc[:,0],autopct=lambda p: '{:.2f}%\n({:1.0f})'.
format(p,(p/100)*sum(df.iloc[:,1])))
    plt.title("5 Most successful countries in 2016 Summer Olympics")
    plt.show()
if __name__=="__main__":
    graph()
    print()
```

5 Most successful countries in 2016 Summer Olympics



11. Write a Python program to draw a scatter graph taking a random distribution in X and Y and plotted against each other.



In [106]:

```
import matplotlib.pyplot as plt
import random
def graph():
    ax=plt.figure(dpi=120).gca()
    ax.tick_params(direction="in",top=True,right=True,bottom=True)
    lim=random.randint(100,150)
    plt.axis([-3,3,-3,3])
    x=[random.uniform(-3,3) for i in range(lim)]
    y=[random.uniform(-3,3) for i in range(lim)]
    plt.scatter(x,y,c='r',s=5)
    plt.title("Scatter Plot")
    plt.xlabel("X")
    plt.ylabel("Y")
    plt.show()
if __name__=="__main__":
    graph()
```

