In [4]:
```
graph={"a" : ["c"],
      'b':['c','e'],
       'c':['a','b','d','e'],
       'd':['c'],
       'e':['c','b'],
       'f':[],
      'g':[]}

def generate_edges(z):
    edges=[]
    for node in z:
        for neighbour in z[node]:
            edges.append((node,neighbour))
    return edges
print(generate_edges(graph))

def find_isolated_nodes(z):
    #retun a list od isolated nodes
    isolated=[]
    for node in z:
        if not z[node]:
            isolated += node
    return isolated

print(find_isolated_nodes(graph))
```

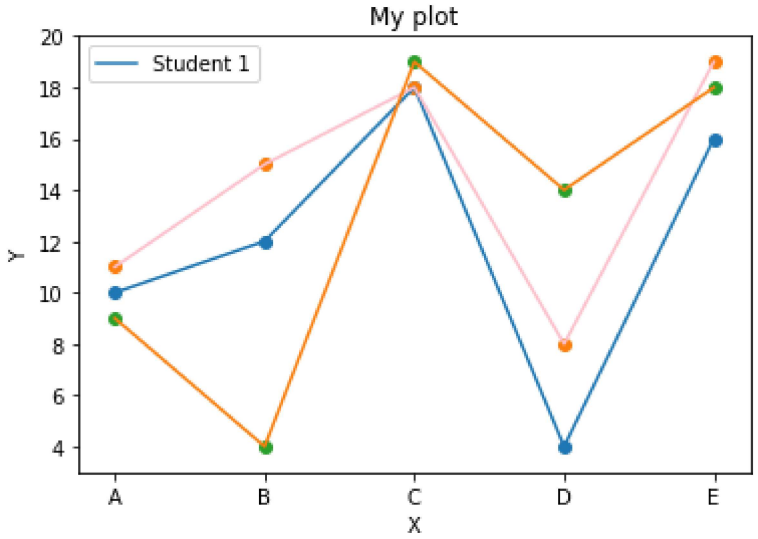
```
[('a', 'c'), ('b', 'c'), ('b', 'e'), ('c', 'a'), ('c', 'b'), ('c', 'd'),
('c', 'e'), ('d', 'c'), ('e', 'c'), ('e', 'b')]
['f', 'g']
```

Visit matplotlib.org for more details

Matplotlib

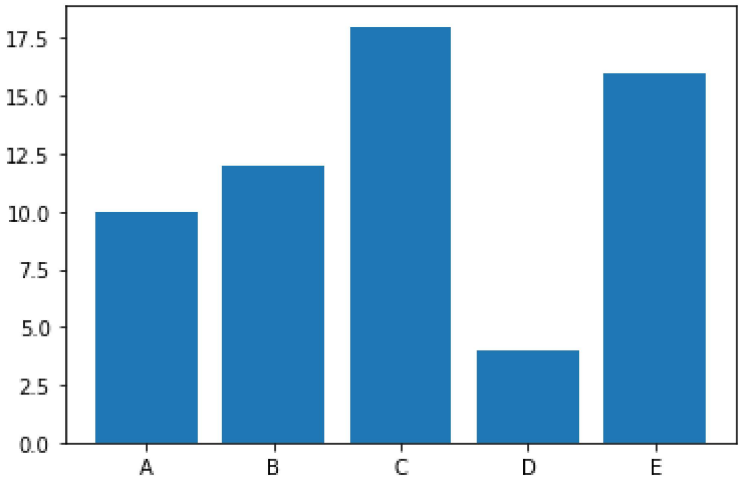
Line and Scatter

In [6]:
```
import matplotlib.pyplot as plt
Division=["A","B","C","D","E"]
student1=[10,12,18,4,16]
student2=[11,15,18,8,19]
student3=[9,4,19,14,18]
plt.plot(Division,student1,label="Student 1")
#line plot
#label
plt.plot(Division,student2,color="pink")
#color
plt.plot(Division,student3)
plt.scatter(Division,student1)
#scatter
plt.scatter(Division,student2)
plt.scatter(Division,student3)
plt.legend()
#turn legend on
plt.title("My plot")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

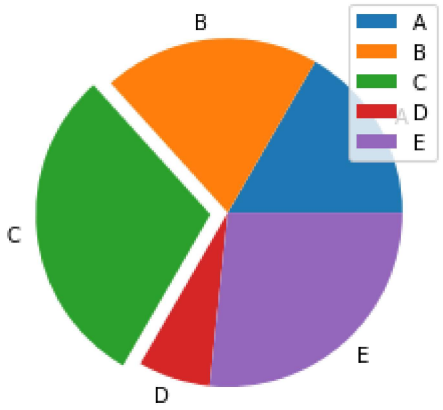


Pie Graph

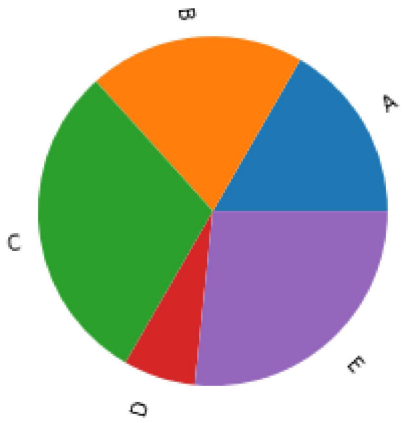
In [17]:
```
plt.bar(Division,student1)
plt.show()
```



In [9]:
```
plt.pie(student1,labels=Division,explode=[0,0,0.1,0,0])
#pie chart
#labels
#taking a pie out(explode)
plt.legend()
#turning on legend
plt.show()
```



In [8]:
```
plt.pie(student1,labels=Division,rotatelabels=True)
#rotating labels
plt.show()
```



Networkx

Empty Graph

In [10]:
```
import networkx as nx

G=nx.Graph()

print(G.nodes())
print(G.edges())

print(type(G.nodes()))
print(type(G.edges()))
```

```
[]
[]
<class 'networkx.classes.reportviews.NodeView'>
<class 'networkx.classes.reportviews.EdgeView'>
```

Adding nodes to graph

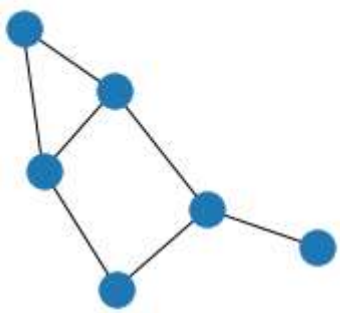
In [14]:
```
import networkx as nx

G=nx.Graph()

# for adding one node
G.add_node("a")

#for adding more than one node pass a list
G.add_nodes_from(["b","c"])

print(G.nodes())
print(G.edges())
```

```
['a', 'b', 'c']
[]
```

Adding eges

In [31]:
```
import networkx as nx
import matplotlib.pyplot as plt

G=nx.Graph()

# for adding one node
##G.add_node("a")

#for adding more than one node pass a list
G.add_nodes_from(["b","c"])
G.add_edges_from([("a","c"),("c","d"),("a",1),(1,"d"),("a",2)])

G.add_edge(1,2)
edge=("d","e")
G.add_edge(*edge)
##edge=("a","b")
##G.add_edge(*edge)


print("Nodes",G.nodes())
print("Edges",G.edges())

nx.draw(G)
#save by plt.savefig("name ")
plt.show()
```

```
Nodes ['b', 'c', 'a', 'd', 1, 2, 'e']
Edges [('c', 'a'), ('c', 'd'), ('a', 1), ('a', 2), ('d', 1), ('d', 'e'), (1,
2)]
```



Renaming Nodes

In [46]:
```
import networkx as nx
import matplotlib.pyplot as plt

G=nx.path_graph(4)

cities=({0:"Toronto", 1:"London",2:"New York", 3:"Berlin"})

H=nx.relabel_nodes(G,cities)
H.add_edge("Toronto","Berlin")
print("Nodes",H.nodes())
print("Edges",H.edges())

nx.draw(H)
#save by plt.savefig("name ")
plt.show()
```

```
Nodes ['Toronto', 'London', 'New York', 'Berlin']
Edges [('Toronto', 'London'), ('Toronto', 'Berlin'), ('London', 'New York'),
('New York', 'Berlin')]
```