Lab 4.2- Graph Data Analytics

In [1]:
```
#!conda install networkx --yes
```

In [2]:
```
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
G = nx.karate_club_graph()
print(nx.info(G))
```

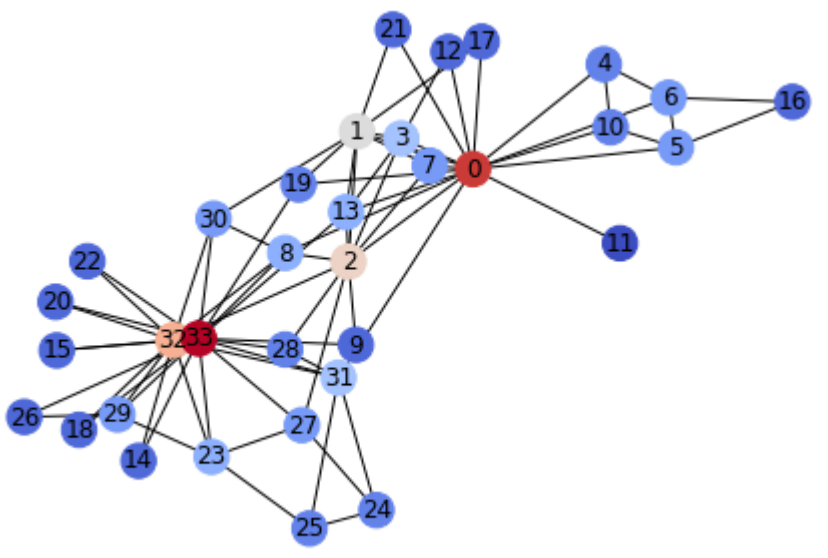
```
Name: Zachary's Karate Club
Type: Graph
Number of nodes: 34
Number of edges: 78
Average degree:   4.5882
```

Calculate page rank by using simple degree centrality

In [3]:
```
def simple_pagerank(G):
    p = np.array([G.degree(index) for node, index in enumerate(G.nodes())])
    return p

values = simple_pagerank(G)
nx.draw(G, cmap=plt.get_cmap('coolwarm'), node_color = values, with_labels=True)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\networkx\drawing\nx_pylab.py:611: MatplotlibDeprecationWar
ning: isinstance(..., numbers.Number)
  if cb.is_numlike(alpha):
```



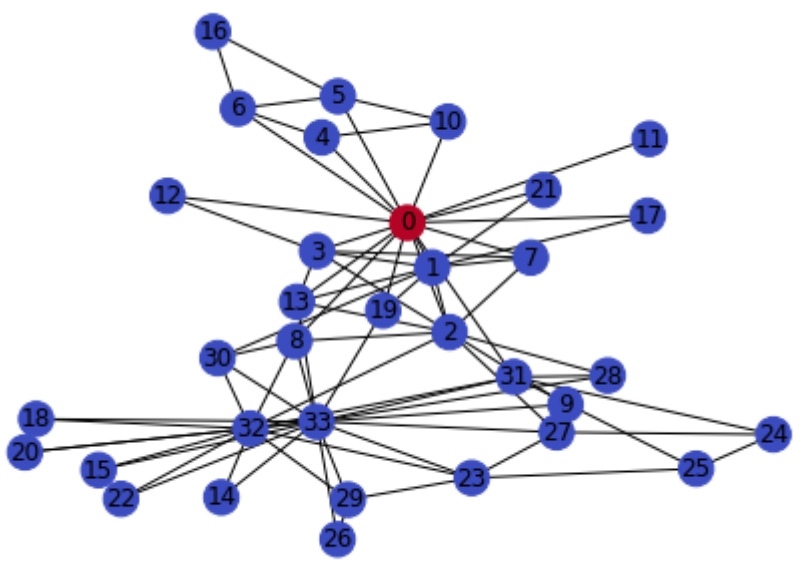
Calculate page rank by updating the centrality of each node with the iteration

In [4]:
```
def pagerank_centrality(G, iter=100):
    p = np.array([1 for i in list(G.nodes())])
    print(p.shape)
    for k in range(iter):
        for i in G.nodes():
            for j in G.nodes():
                # update the centrality
                try:
                    p[i] += int(G.number_of_edges(i, j) * p[j] / G.degree[j])
                except:
                    pass

        norm = sum(p)
        p = p / norm
    return p


values = pagerank_centrality(G)
nx.draw(G, cmap=plt.get_cmap('coolwarm'), node_color = values, with_labels=True)
```

```
(34,)
```



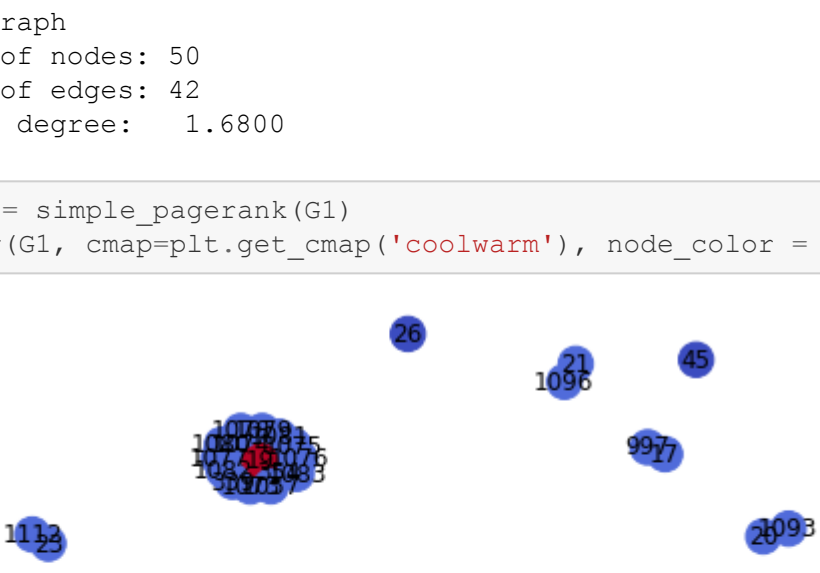
In [5]:
```
G = nx.read_edgelist("gr0.California.edegs.txt")
G.remove_nodes_from(list(nx.isolates(G)))
print(nx.info(G))
G1 = G.subgraph(list(G.nodes())[100:150])
print(nx.info(G1))
```

```
Name:
Type: Graph
Number of nodes: 6175
Number of edges: 15969
Average degree:   5.1721
Name:
Type: Graph
Number of nodes: 50
Number of edges: 42
Average degree:   1.6800
```

In [6]:
```
values = simple_pagerank(G1)
nx.draw(G1, cmap=plt.get_cmap('coolwarm'), node_color = values, with_labels=True)
```


In [7]:
```
values = pagerank_centrality(G1)
nx.draw(G1, cmap=plt.get_cmap('coolwarm'), node_color = values, with_labels=True)
```

```
(50,)
```



In [ ]:
```

```