```
** MyGraphTest1: testing class MyGraph only.
** VertexNames={zero,one,two,three} for Vertices {0,1,2,3}
** EdgeSet={ {0,1},{0,2}, {0,3}, {2,3}, {1,3} } with weights
1,2,3,4,5

Now testing each method:

Testing Constructor...
>>>>>> MyGraph() to-do1 needs to be implemented<<<<<<<<<<
v0:zero
v1:one
v2:two
v3:three
>>>>>> MyGraph() to-do2 needs to be implemented<<<<<<<<<<
e={v0:zero,v1:one} dist=1.0
e={v0:zero,v2:two} dist=2.0
e={v0:zero,v3:three} dist=3.0
e={v0:zero,v1:one} dist=1.0
e={v1:one,v3:three} dist=5.0
e={v0:zero,v2:two} dist=2.0
e={v2:two,v3:three} dist=4.0
e={v0:zero,v3:three} dist=3.0
e={v1:one,v3:three} dist=5.0
e={v2:two,v3:three} dist=4.0
Completed Constructor.

numVertices()=4
numEdges()=5
vertices() and show indices:0,1,2,3,
edges() and show weights:1.0,2.0,3.0,4.0,5.0,

degree of vertex 0 =3
incidentEdges(0):  e={v0:zero,v1:one} dist=1.0; e={v0:zero,v2:two}
dist=2.0; e={v0:zero,v3:three} dist=3.0;
degree of vertex 1 =2
incidentEdges(1):  e={v0:zero,v1:one} dist=1.0; e={v1:one,v3:three}
dist=5.0;
degree of vertex 2 =2
incidentEdges(2):  e={v0:zero,v2:two} dist=2.0; e={v2:two,v3:three}
dist=4.0;
degree of vertex 3 =3
incidentEdges(3):  e={v0:zero,v3:three} dist=3.0;
e={v1:one,v3:three} dist=5.0; e={v2:two,v3:three} dist=4.0;

Testing getEdge(v,w) for each pair of vertices (v,w) and showing
distances:
dist(0,1)=1.0; dist(0,2)=2.0; dist(0,3)=3.0;
dist(1,0)=1.0; dist(1,3)=5.0;
dist(2,0)=2.0; dist(2,3)=4.0;
dist(3,0)=3.0; dist(3,1)=5.0; dist(3,2)=4.0;

Testing: endVertices, opposite
e={v0:zero,v1:one} dist=1.0: ends are 0,1 ,opposite of 0 is 1,
opposite of 1 is 0
```

```
e={v0:zero,v2:two} dist=2.0: ends are 0,2 ,opposite of 0 is 2,
opposite of 2 is 0
e={v0:zero,v3:three} dist=3.0: ends are 0,3 ,opposite of 0 is 3,
opposite of 3 is 0
e={v2:two,v3:three} dist=4.0: ends are 2,3 ,opposite of 2 is 3,
opposite of 3 is 2
e={v1:one,v3:three} dist=5.0: ends are 1,3 ,opposite of 1 is 3,
opposite of 3 is 1

**End Test !!!
```