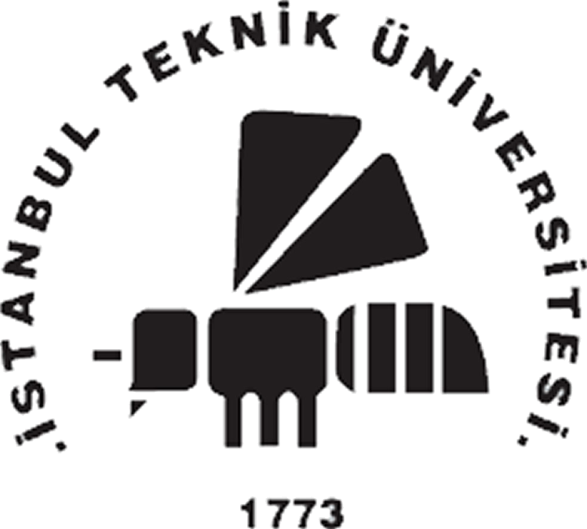
**I.T.U.**

**Faculty of Electric-Electronic**

**Computer Engineering**



Lesson name: Object Oriented Programming

Lesson Code: BLG252E

Name Surname: Aykut Akın

Number: 040080177

Instructor’s Name: Feza Buzluca

Due Date: 14.03.2010

**Introduction**

I implemented classes that can be used to read and represent undirected graphs in C++.

My classes represent graphs using its name and an array of nodes. Also each node in a graph

hold references to the adjacent nodes in the graph.

In my homework there exists a has-a relationship between a graph class and node class. Each

node may exist in a single graph but a graph can contain many exposing a one-to-many relationship.

A graph can be created in one of the three ways.

– An empty graph that doesn't contain any node or edge

– A graph with a predetermined size and arbitrary node names, but no connections.

– A graph as a copy of other

These operations can be made for graph

– A node can be added to a graph. There can't be two nodes with same name in a graph.

– A node can be deleted from a graph, together with its connections.

– An edge can be added/deleted from a graph. An edge can be added/deleted only if its nodes

exist in the graph.

– The graph can be output to screen

– Two graphs can be intersected. The resulting graph contains the intersection of both nodes and

edges.

– Two graphs can be unified. The resulting graph contains the union of both nodes and edges.

**Explanation of Graph.h**

numNodes: Number of nodes included in the graph

numEdges: Number of edges included in the graph

name: Name of the graph

nodes: Nodes which are included in the graph

graphCount: Total graph count

Graph(): Constructor

Graph(int): Constructor which takes a number of nodes as a parameter

Graph(const Graph &): Copy constructor

~Graph(): Destructor

newNode(string): A function that adds a new node to the graph and takes a name of node as a

parameter. Returns true if the node is added

newEdge(string,string): A function that adds a new edge between two nodes and takes the names

of the nodes as a parameter. Returns true if the edge is added

deleteNode(string): A function that deletes a node from the graph and takes the name of the

node as a parameter. Returns null if the node is absent

deleteEdge(string,string): A function that deletes an edge between two nodes and takes the name

of the nodes as a parameter. Returns false if the edge is absent

printGraph(): A function that print the graph to the screen

intersect(const Graph &): A function that intersect the two graphs and takes a graph as a

parameter. Prints a new intersection graph to the screen

unite(const Graph &):A function that unites the two graphs and takes a graph as a

parameter. Prints a new union graph to the screen

**Explanation of Node.h**

string name: Name of the node

adjacent: Pointer to adjacent of node

Node(): Constructor

Node(string): Constructor which takes a name of the node as a parameter

Node(const Node &): Copy constructor

~Node(): Destructor

delAdjacent(string): A function that deletes the adjacent of the node and takes the name of the

node as a parameter

addAdjacent(const Node &): A function that adds a new adjacent to the node and takes the

node as a parameter

getAdjacents(): A function that returns the adjacent of node

getName(): A function that returns the name of node

numEdges: Number of edges that the node has

**Missing parts of the homework**

Having its content from a file: You need to create graphs with your hands. You can not include the

graph from a file.

Each graph must have a graph file attached to it: In my homework you can only see the output on the

screen. There are no such files for the graphs that you created.