



DOCUMENTATION

J.C.Lakmal
2017/CS/092

Content

1. Introduction.....	3
2. Overview.....	4
3. Assumptions.....	5
4. Setup System.....	6
5. Design.....	7

Introduction

My main goal is develop the user friendly graph implementation system. According to this goal, I started to implement the graph. I used to implement this graph netbeans ide and jdk 10.

Overview

I used netbeans ide , then I created a project. Then I created some packages in the project. I created some classes inside the packages. Then I used object oriented programming concept to implement this graph project. So, any user can easily to understand my program and I can search any errors easily. This graph system I am not use to any jframe form because I thought It is not very useful this system. Then I got all the inputs Using scanner class.

Assumptions

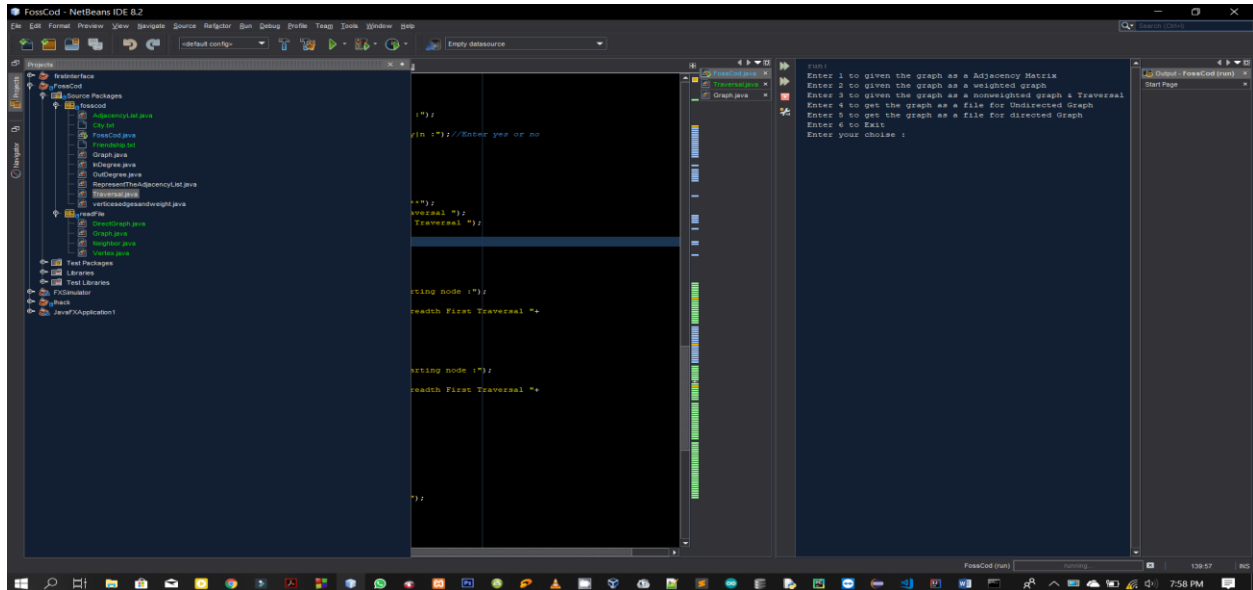
My main Assumption is only user can give the inputs using keyboard.

Setup System

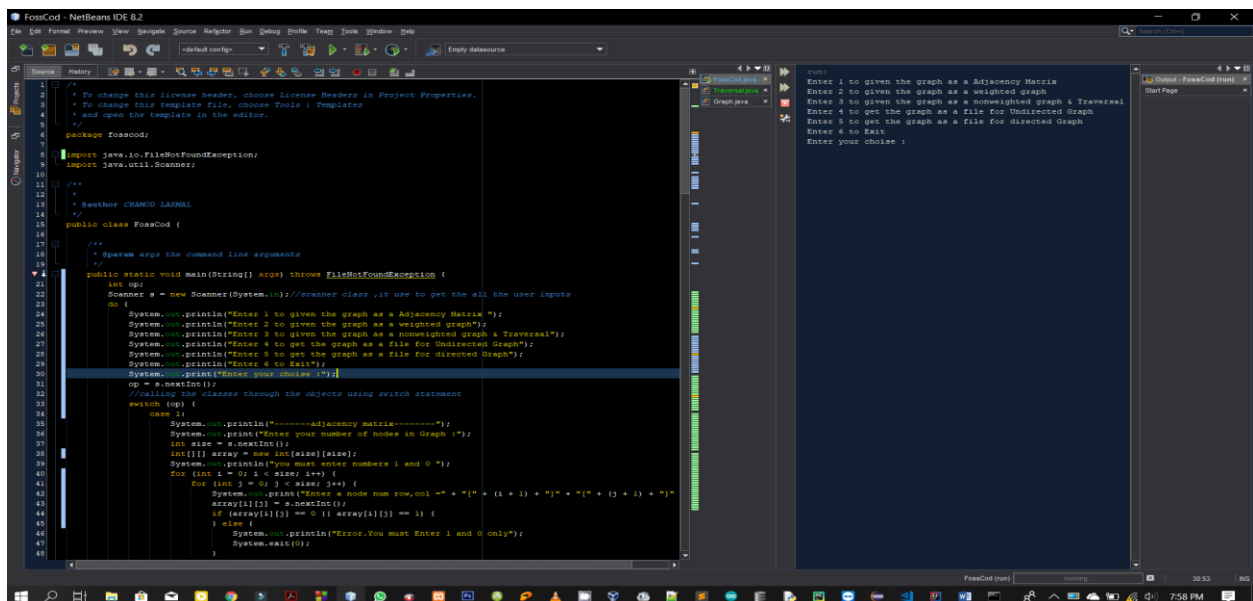
You should clone the my project from Github. Then you should open it on netbeans ide.

Design

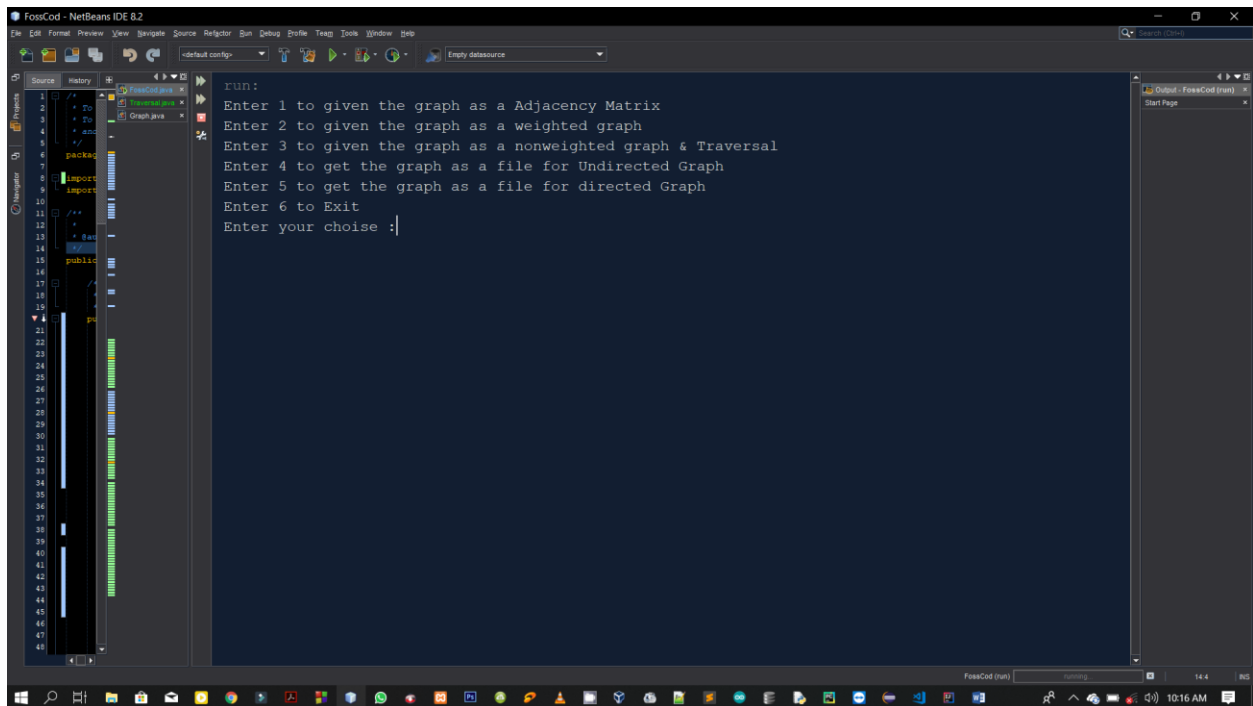
First you can see this window.



Then you want to open the FossCod.java (main) class. Then you can see this code.



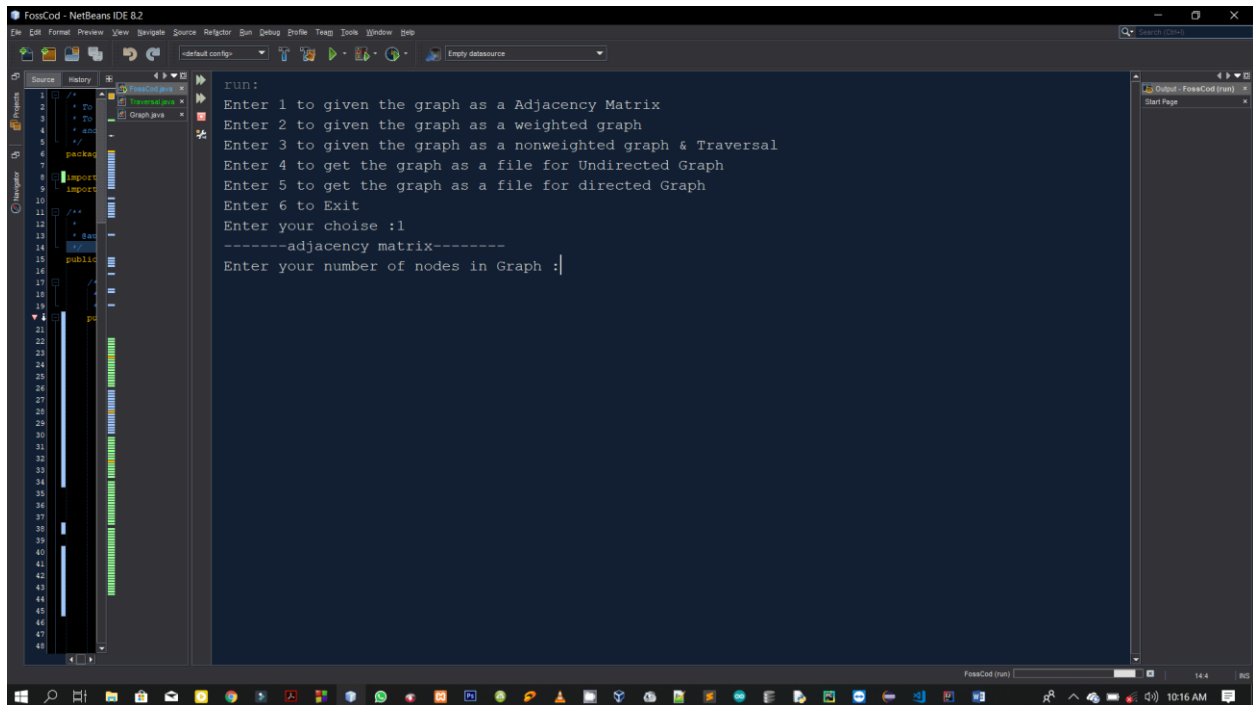
Then you can run it (shift+ F6). Now you can see this.



The screenshot shows the FossCod IDE interface. The main editor window displays a Java program with a menu for graph operations. The menu options are: Enter 1 to given the graph as a Adjacency Matrix, Enter 2 to given the graph as a weighted graph, Enter 3 to given the graph as a nonweighted graph & Traversal, Enter 4 to get the graph as a file for Undirected Graph, Enter 5 to get the graph as a file for directed Graph, Enter 6 to Exit, and Enter your choise :|. The left sidebar shows the project structure with 'FossCod.java' and 'Graph.java' files. The right sidebar shows the 'Output - FossCod (run)' window, which is currently empty.

```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choise :|
```

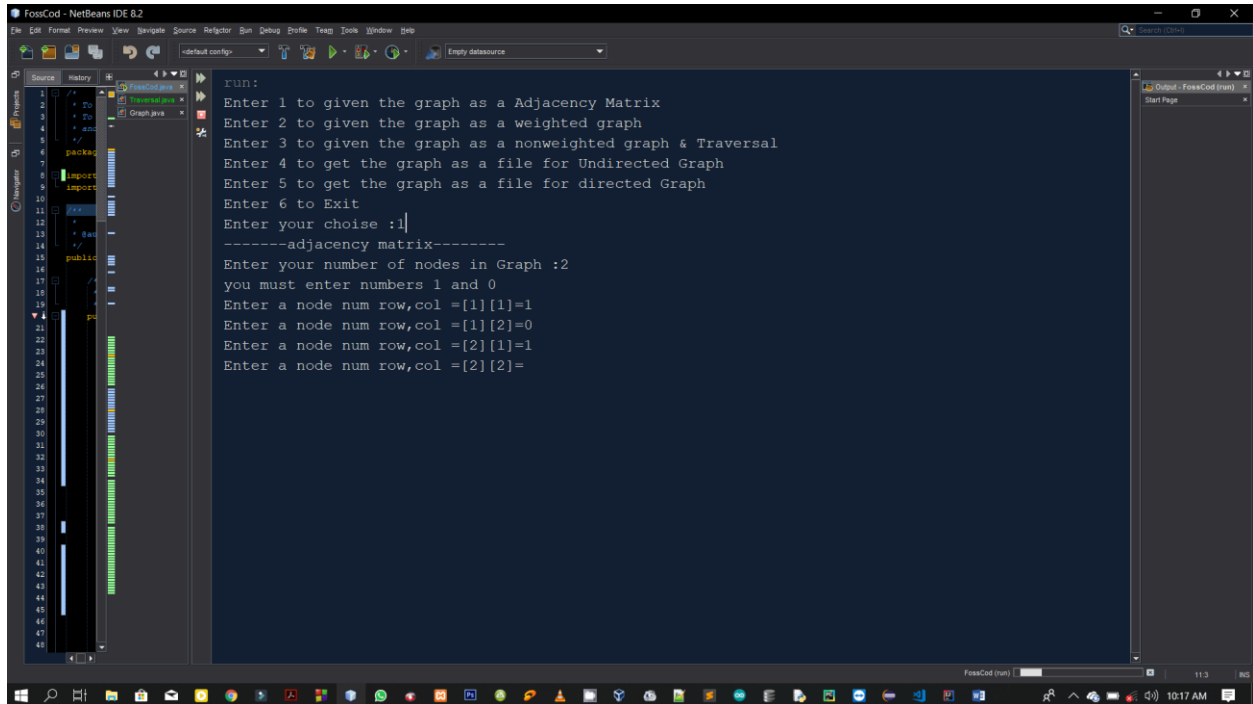
Then press 1 if you want. Then you can see this ,



The screenshot shows the FossCod IDE interface after the user has pressed '1'. The main editor window displays the same menu as before, but the prompt 'Enter your choise :|' has been replaced with '-----adjacency matrix-----' and 'Enter your number of nodes in Graph :|'. The left sidebar shows the project structure with 'FossCod.java' and 'Graph.java' files. The right sidebar shows the 'Output - FossCod (run)' window, which is currently empty.

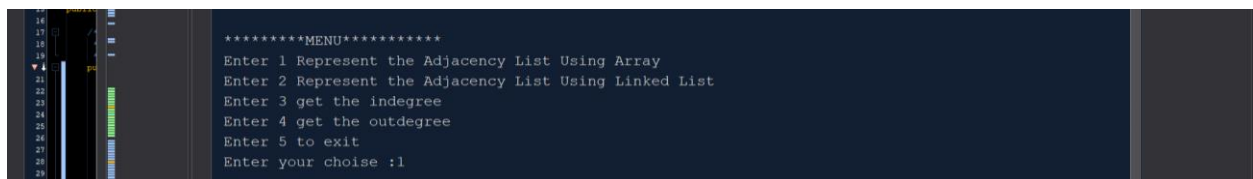
```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choise :1
-----adjacency matrix-----
Enter your number of nodes in Graph :|
```


Then you can enter number of nodes you want. I give it 2, then It creates the 2*2 matrix. It is the Adjacency matrix. Now you can give your graph as adjacency matrix.



```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choise :1
-----adjacency matrix-----
Enter your number of nodes in Graph :2
you must enter numbers 1 and 0
Enter a node num row,col =[1][1]=1
Enter a node num row,col =[1][2]=0
Enter a node num row,col =[2][1]=1
Enter a node num row,col =[2][2]=
```

After entering your matrix you can see this menu.



```
*****MENU*****
Enter 1 Represent the Adjacency List Using Array
Enter 2 Represent the Adjacency List Using Linked List
Enter 3 get the indegree
Enter 4 get the outdegree
Enter 5 to exit
Enter your choise :1
```

Then you can get 4 operations to do with this adjacency matrix.

- Press 1 to get your graph as adjacency list
This time I create the array then you can see the adjacency list using array concept. Its disadvantage is It waste lot of memory.

```
*****MENU*****
Enter 1 Represent the Adjacency List Using Array
Enter 2 Represent the Adjacency List Using Linked List
Enter 3 get the indegree
Enter 4 get the outdegree
Enter 5 to exit
Enter your choice :1

-----Adjacency List-----
[1]-->[1]
[2]-->[1]-->[2]
```

- Press 2 to get your graph as adjacency list(Using linked list)
This time I create the linked list then you can see the adjacency list using linked list concept. Its advantage is It save the memory.

```
*****MENU*****
Enter 1 Represent the Adjacency List Using Array
Enter 2 Represent the Adjacency List Using Linked List
Enter 3 get the indegree
Enter 4 get the outdegree
Enter 5 to exit
Enter your choice :2
0==>[1]
1==>[1, 2]
```

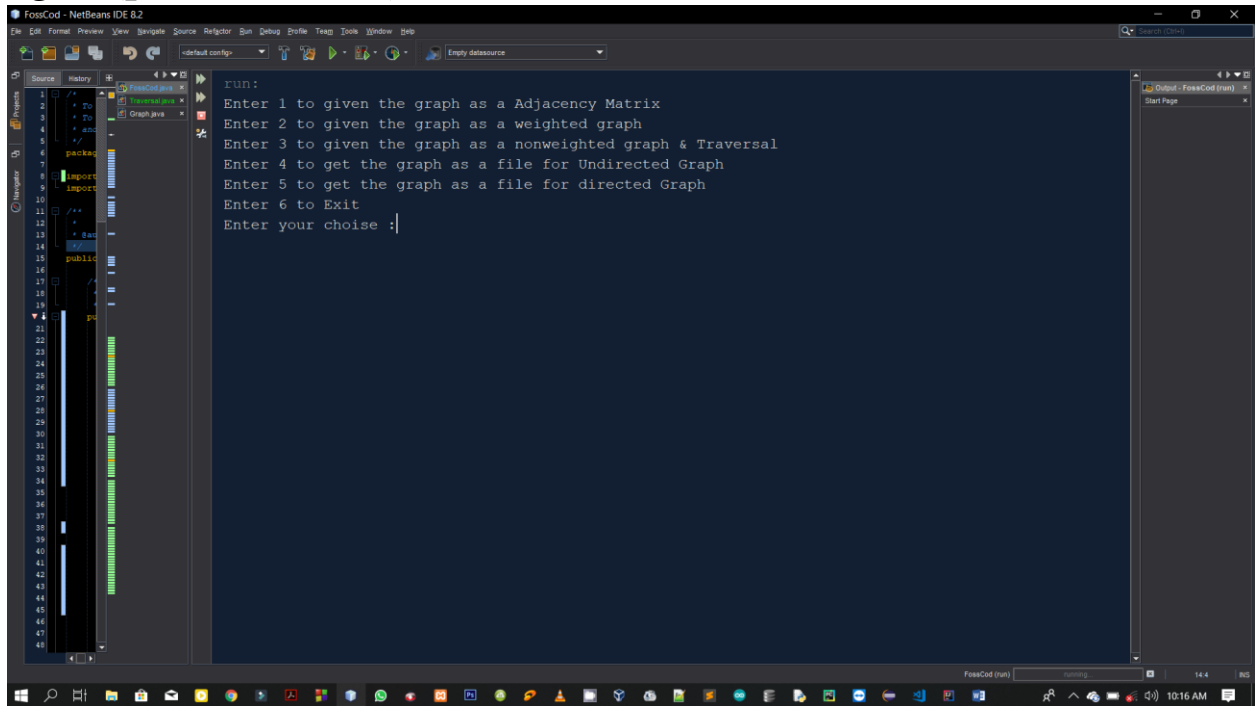
- Press 3 to get idegree. Then you want to enter the node you want to get the Indegree.

```
*****MENU*****
Enter 1 Represent the Adjacency List Using Array
Enter 2 Represent the Adjacency List Using Linked List
Enter 3 get the indegree
Enter 4 get the outdegree
Enter 5 to exit
Enter your choice :3
Enter your node, calculate the indegree :1
Number 1 Indegree is :2
```

- Press 4 to get the outdegree. Then you want to enter the node you want to get the Outdegree.

```
*****MENU*****
Enter 1 Represent the Adjacency List Using Array
Enter 2 Represent the Adjacency List Using Linked List
Enter 3 get the indegree
Enter 4 get the outdegree
Enter 5 to exit
Enter your choice :4
Enter your node, calculate the Outdegree :2
Number 2 Outdegree is :2
```

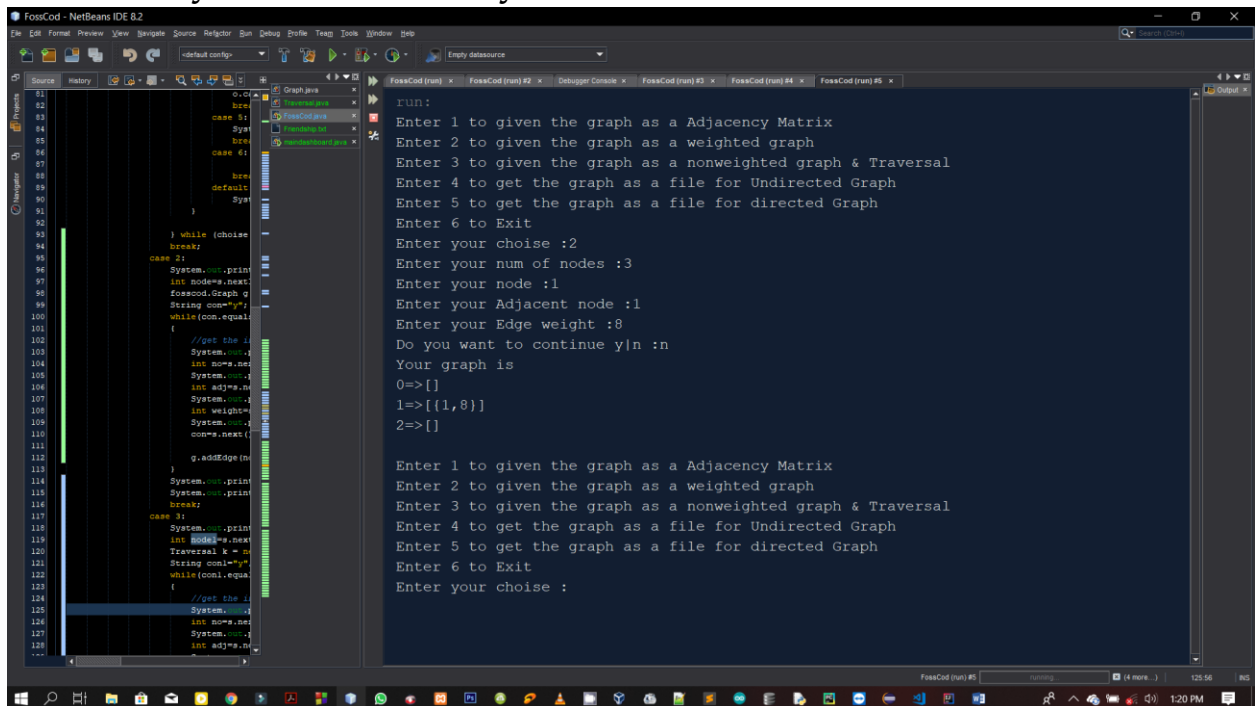
Then you can exit the program , then you can run the code again (press shift+F6).



The screenshot shows the NetBeans IDE with the FossCod.java file open. The code defines a menu for graph operations. The output window shows the program running and displaying the menu options.

```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :|
```

Enter 2 If you want. Then you can see this.



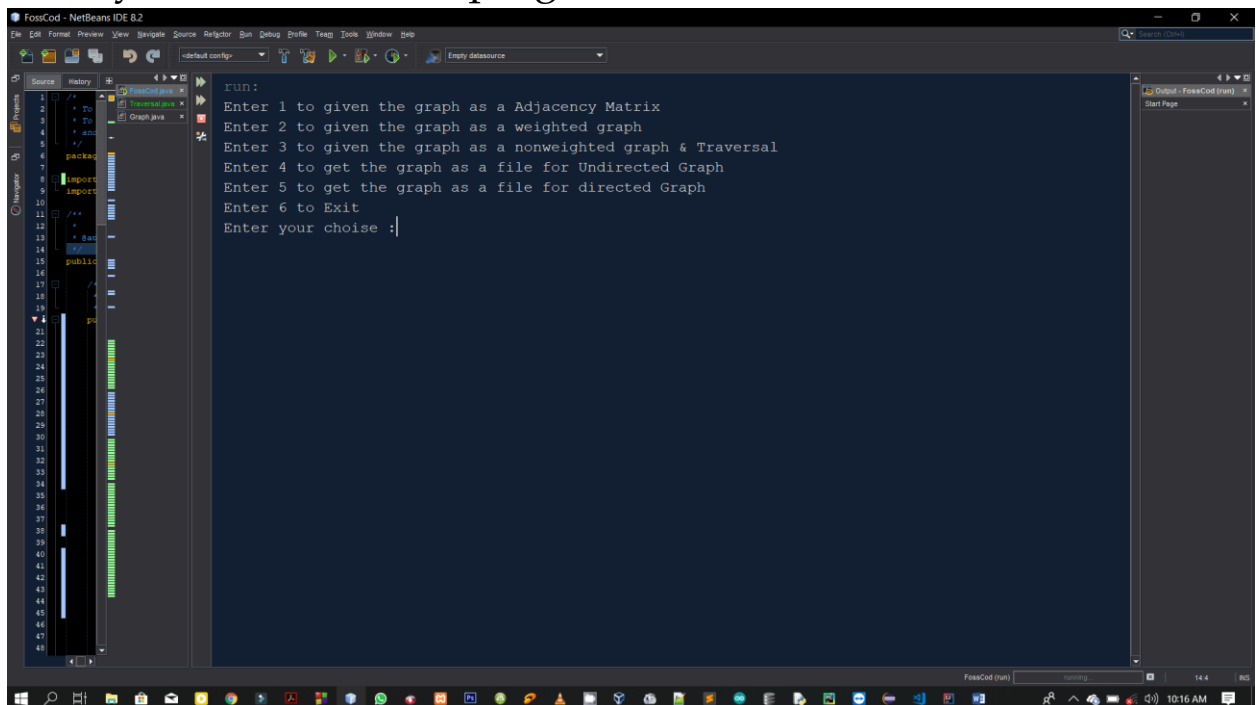
The screenshot shows the NetBeans IDE with the FossCod.java file open. The code defines a menu for graph operations. The output window shows the program running and displaying the menu options. The user has entered 2, and the program has prompted for the number of nodes, the node, the adjacent node, and the edge weight. The graph data structure is shown as an adjacency list.

```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :2
Enter your num of nodes :3
Enter your node :1
Enter your Adjacent node :1
Enter your Edge weight :8
Do you want to continue y/n :n
Your graph is
0=>[]
1=>[[1,8]]
2=>[]

Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :
```

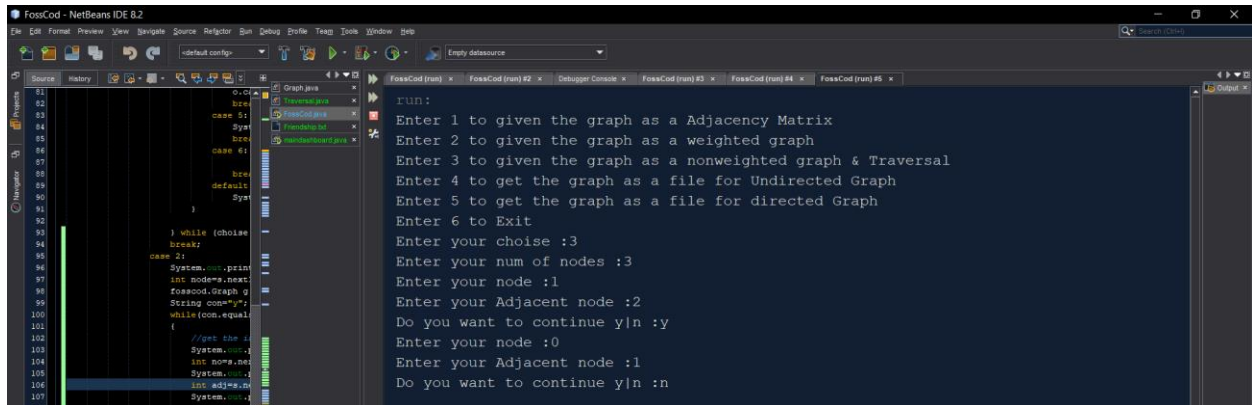
Then you can input inputs as I input above. You can input any number of inputs to the graph. Because I use the while loop. If you press the “n” then you can finish the input given.

Now you can exit the program pressing 6.
Then you can re run the program.



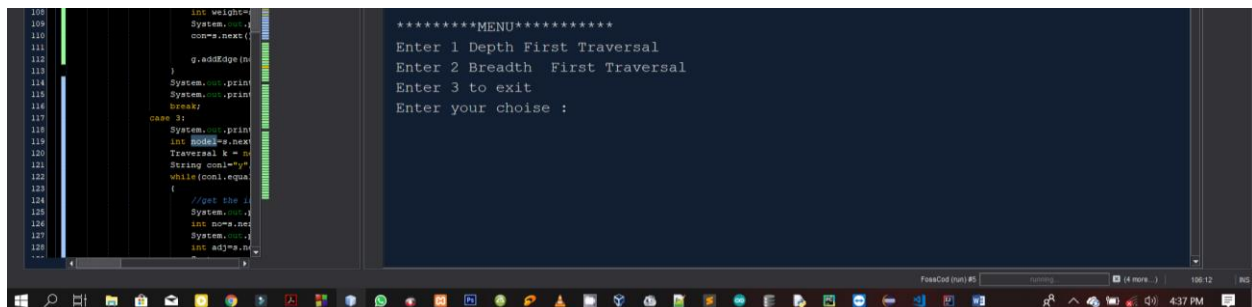
```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choise :|
```

Then you can enter 3 if you want. Then you can enter nodes as I enter below.



```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :3
Enter your num of nodes :3
Enter your node :1
Enter your Adjacent node :2
Do you want to continue y/n :y
Enter your node :0
Enter your Adjacent node :1
Do you want to continue y/n :n
```

After entering nodes you can see this.

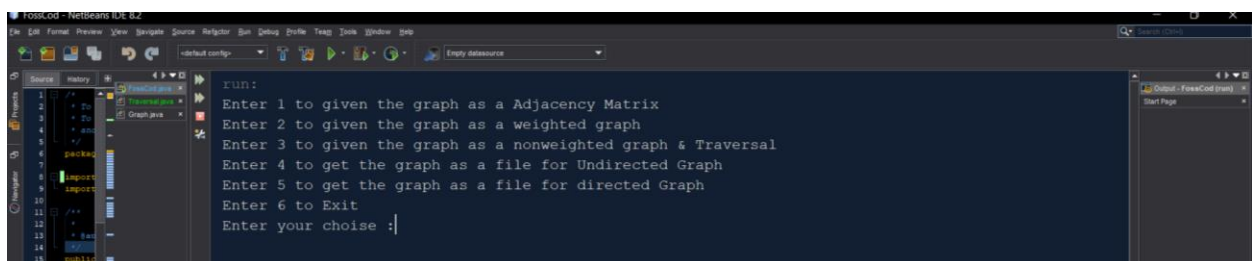


```
*****MENU*****
Enter 1 Depth First Traversal
Enter 2 Breadth First Traversal
Enter 3 to exit
Enter your choice :
```

Now you can Traverse the Graph. Enter 1 for DFS , and Enter 2 for BFS.

Enter 3 to exit.

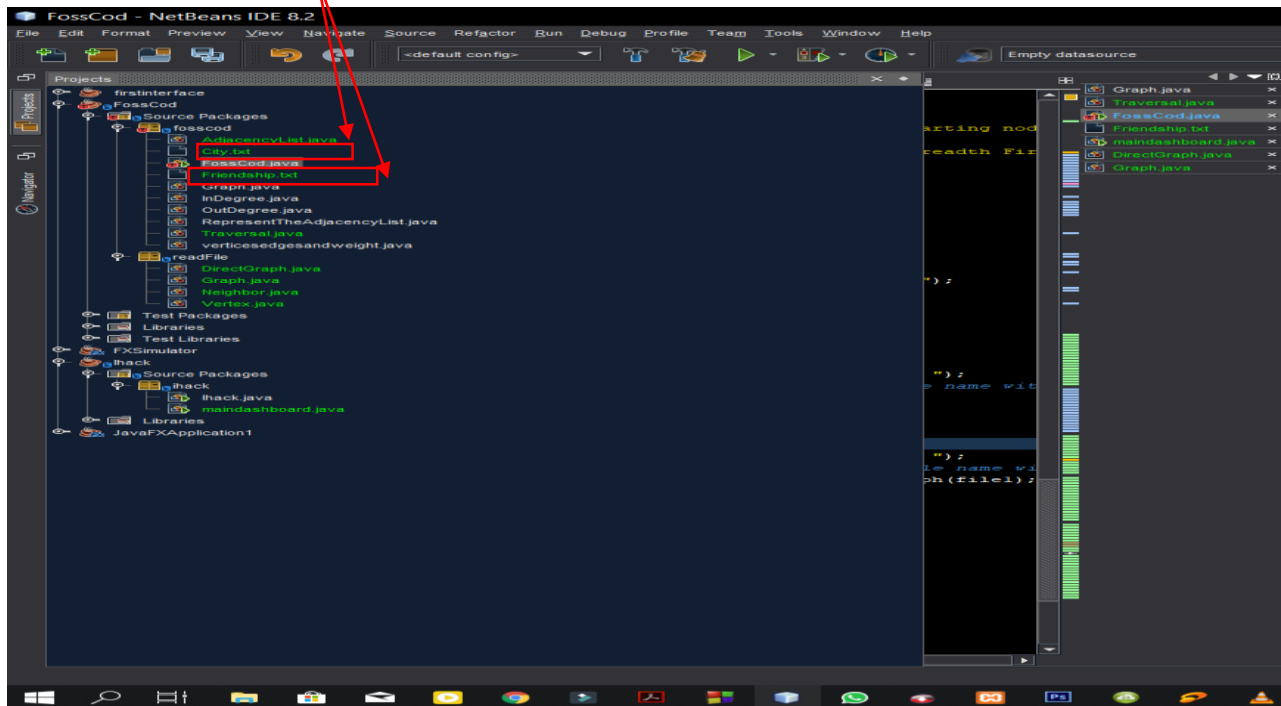
After that ,



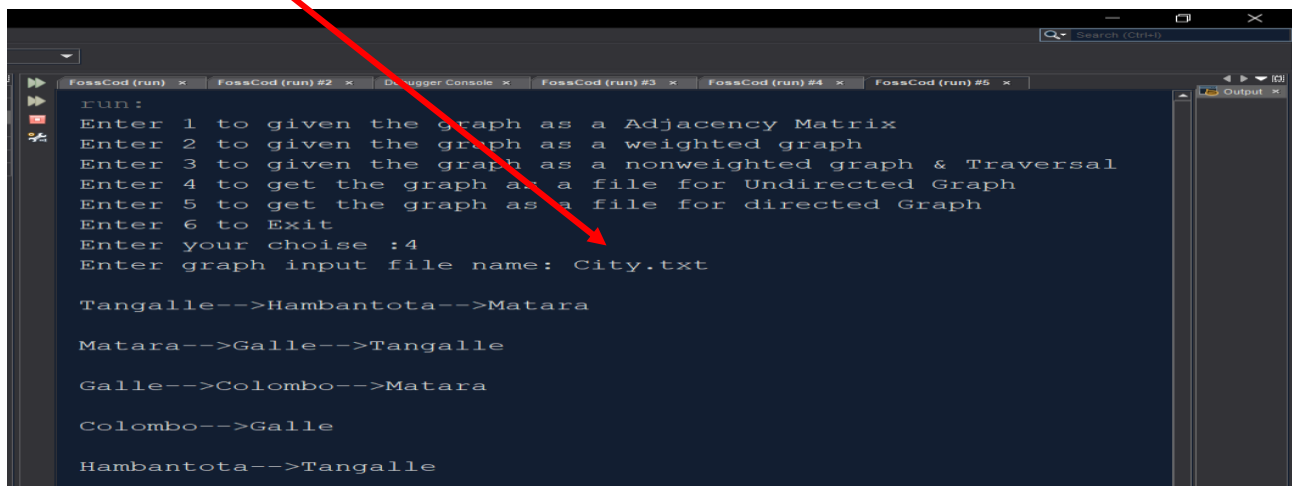
```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :|
```

Enter 4 to get the graph as a file. If you enter 4,

Before entering the file name you should import the file into your project. As below,

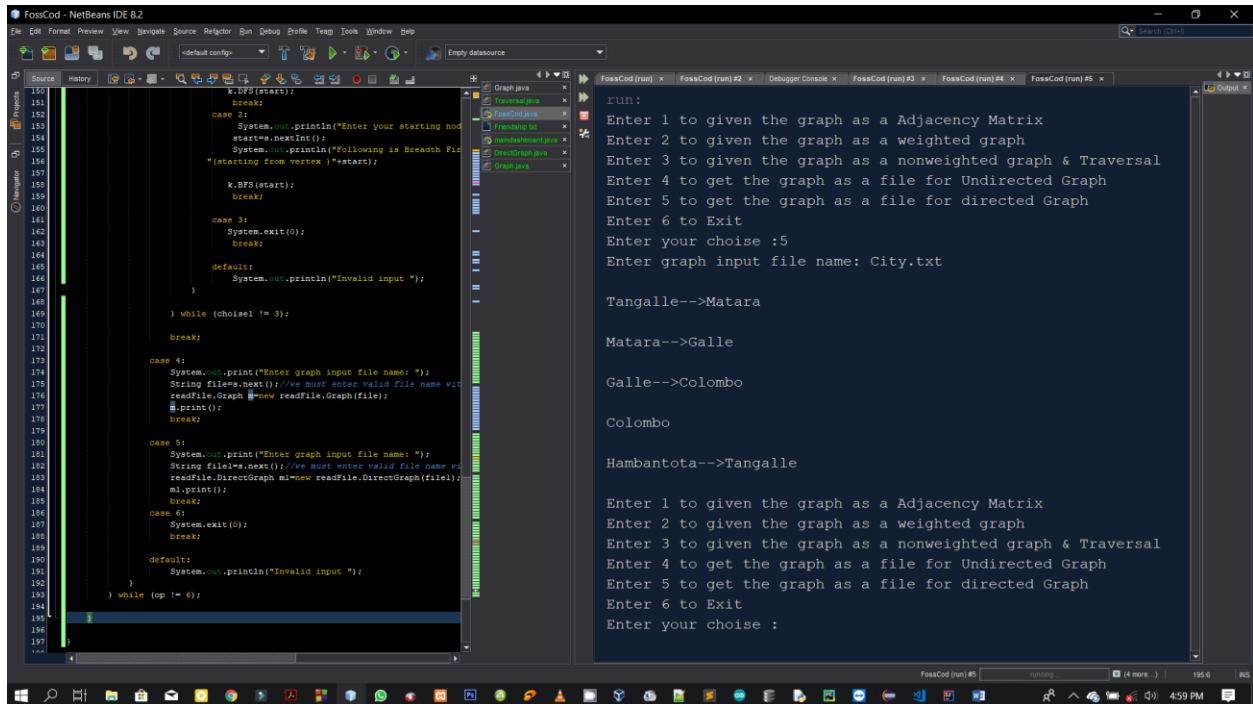


Now you can enter the file name with extension. As below.



This time you can get the graph as an Undirected graph.

If you enter 5 , you can get the graph as a directed graph. As below,



The screenshot displays the FoxCod IDE interface. The left pane shows the source code of a Java application. The right pane shows the execution output, which includes a menu of options for graph operations and a list of directed edges from a file named 'City.txt'.

```
151:         k.DFS(start);
152:         break;
153:     case 2:
154:         System.out.println("Enter your starting node");
155:         startw.nextInt();
156:         System.out.println("Following is Breadth First Search starting from vertex "+startw);
157:         k.BFS(start);
158:         break;
159:     case 3:
160:         System.exit(0);
161:         break;
162:     default:
163:         System.out.println("Invalid input ");
164:     }
165: } while (choisel != 3);
166: break;
167:
168: case 4:
169:     System.out.println("Enter graph input file name: ");
170:     String filew.next();//we must enter valid file name vlt
171:     readfile.Graph g=new readfile.Graph(filew);
172:     g.print();
173:     break;
174:
175: case 5:
176:     System.out.println("Enter graph input file name: ");
177:     String filew.next();//we must enter valid file name vlt
178:     readfile.DirectGraph g1=new readfile.DirectGraph(filew);
179:     g1.print();
180:     break;
181:
182: case 6:
183:     System.exit(0);
184:     break;
185:
186: default:
187:     System.out.println("Invalid input ");
188: }
189: } while (op != 6);
190: }
```

```
run:
Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :5
Enter graph input file name: City.txt

Tangalle-->Matara
Matara-->Galle
Galle-->Colombo
Colombo
Hambantota-->Tangalle

Enter 1 to given the graph as a Adjacency Matrix
Enter 2 to given the graph as a weighted graph
Enter 3 to given the graph as a nonweighted graph & Traversal
Enter 4 to get the graph as a file for Undirected Graph
Enter 5 to get the graph as a file for directed Graph
Enter 6 to Exit
Enter your choice :
```

This is the all of my project. I think you can understand above my description.

Thank you