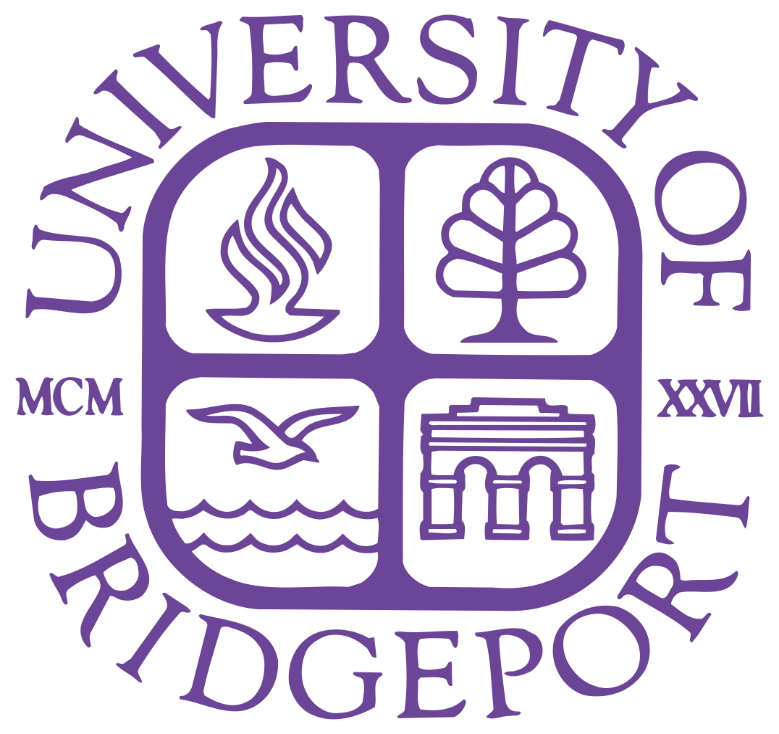
UNIVERSITY OF BRIDGEPORT

FALL-2017



CPSC-501-11- OOP with Design Patterns-2017FA

Assignment-2

By:

Jagrithi Palachanda Chittiappa

ID: 1033162

Due date- 10/09/2017

Submitted date- 10/09/2017

**Table of Contents**

Documented Source Code 3

Screenshots 14

Things you learned 16

Work Cited 16

**Header file:**

Filename: Navigator.h

Author: Jagrithi

Date: 10/09/2017

Compiler: Netbeans 8.2

Description: This file contains all the members and the member method declarations.

#ifndef NAVIGATOR\_H

#define NAVIGATOR\_H

#include <iostream>

#include <stdio.h>

#include <fstream>

#include <stdlib.h>

using namespace std;

class Navigator

{

private:

int row,col,r,c,srow,scol,erow,ecol;

char \*\*a;

int len,count;

string array1[50];

bool flag;

int dummy;

public:

Navigator();

void Read(string);

void matrix();

void move();

void check();

void Right();

void Up();

void Left();

void Down();

void display();

void deallocate();

friend ostream &operator<< (ostream &,Navigator &);

friend void operator>> (ifstream &,Navigator &);

};

#endif

**Source file**

File name: Navigator.cpp

Author: Jagrithi

Date: 10/09/2017

Compiler: Netbeans 8.2

Description: This has all the definitions of the methods used in the program.

#include <iostream>

#include <stdio.h>

#include <fstream>

#include <string>

#include <stdlib.h>

#include <ctype.h>

#include <time.h>

#include "Navigator.h"

using namespace std;

Navigator::Navigator()

{

row=0;

col=0;

count=0;

srow=0;

scol=0;

erow=0;

ecol=0;

r=0;c=0;

flag=0;

len=0;

}

/\*Navigator::~Navigator()

{

for (int i = 0; i < row; i++)

delete [] a[i] ;

delete []a;

}\*/

void Navigator::matrix()

{

a = new char \*[row];

r=row;

for (int i = 0; i < row; i++)

a[i] = new char[col];

c=col;

for(int i=1;i<r;i++)

for(int j=1;j<c;j++)

a[i][j]='=';

}

void operator>>(ifstream &file1, Navigator &drive)

{

if(file1.is\_open())

{

while(!file1.eof()) //file is open

{

getline(file1,drive.array1[drive.len]); //eof hasn't been reached yet

drive.len++;

} //all the contents are read and is stored in array1

}

file1.close(); //file is closed

int i=0;

do

{

drive.row=drive.array1[i][1]-48;

drive.col=drive.array1[i][3]-48;

if(drive.count==0)

{

drive.row=drive.row+1;

drive.col=drive.col+1;

drive.matrix();

}

if (drive.count==1)

{

drive.a[drive.row][drive.col]='C';

drive.srow=drive.row;

drive.scol=drive.col;

}

if(drive.count==2)

{

drive.erow=drive.row;

drive.ecol=drive.col;

drive.a[drive.erow][drive.ecol]='E';

}

if(drive.count>=3)

{

if(drive.array1[i]=="-1"){

drive.display();

cout<<endl<<"Navigation begins"<<endl;

drive.move();

drive.count=-1;

drive.deallocate();

}

else

drive.a[drive.row][drive.col]='O';

}

drive.count++;

i++;

}while(i<drive.len);

}

void Navigator::display()

{

for(int i=1;i<r;i++)

{

for(int j=1;j<c;j++)

{

cout<<a[i][j];

cout<<" ";

}

cout<<endl;

}

}

void Navigator::move()

{

check();

}

void Navigator::check()

{

while(!((srow==erow) && (scol==ecol)))

{

if((a[srow-1][scol]!='O') && (srow>erow))

{

Up();

}

else if((a[srow][scol+1]!='O') && (scol<ecol))

{

Right();

}

else if((a[srow][scol-1]!='O') && (scol>ecol))

{

Left();

}

else if((a[srow+1][scol]!='O') && (srow<erow))

{

Down();

}

if(srow==erow && scol==ecol)

{

if(a[srow][scol]==a[erow][ecol])

cout<<"Car has reached the end point"<<endl;

}

}

}

void Navigator::Up()

{

if(a[srow-1][scol]!='O')

{

cout<<"Move direction: Up"<<endl;

srow--;

a[srow][scol]='C';

a[srow+1][scol]='=';

display();

}

}

void Navigator::Right()

{

if(a[srow][scol+1]!='O')

{

cout<<"Move direction: Right"<<endl;

scol++;

a[srow][scol]='C';

a[srow][scol-1]='=';

display();

}

}

void Navigator::Left()

{

if(a[srow][scol-1]!='O')

{

cout<<"Move direction: Left"<<endl;

scol--;

a[srow][scol]='C';

a[srow][scol+1]='=';

display();

}

}

void Navigator::Down()

{

if(a[srow+1][scol]!='O')

{

cout<<"Move direction: Down"<<endl;

srow++;

a[srow][scol]='C';

a[srow-1][scol]='=';

display();

}

}

ostream &operator<<(ostream &file2,Navigator &drive)

{

for(int i=1;i<drive.r;i++)

{

for(int j=1;j<drive.c;j++)

{

file2<<drive.a[i][j];

file2<<" ";

}

file2<<endl;

}

return file2;

}

void Navigator::deallocate(){

for(int i=1;i<row;i++){

delete []a[i];

}

delete[]a;

}

**Driver program:**

File name: test.cpp

Author: Jagrithi

Date: 10/09/2017

Compiler used: Netbeans 8.2

Description: This has the main function and the objects of the class.

#include<iostream>

#include "Navigator.h"

using namespace std;

int main()

{

Navigator drive;

ifstream file1;

file1.open("Input.txt");

file1>>drive;

cout<<"The final map is:"<<endl;

cout<<drive;

return 0;

}**Input files:**

1. Filename: Input.txt

Author: Jagrithi

Date: 10/09/2017

Compiler used: GNU

Description: This is the main input file which is fed to the program.

(6,5)

(1,1)

(3,5)

(2,1)

(1,3)

(2,4)

-1

(5,5)

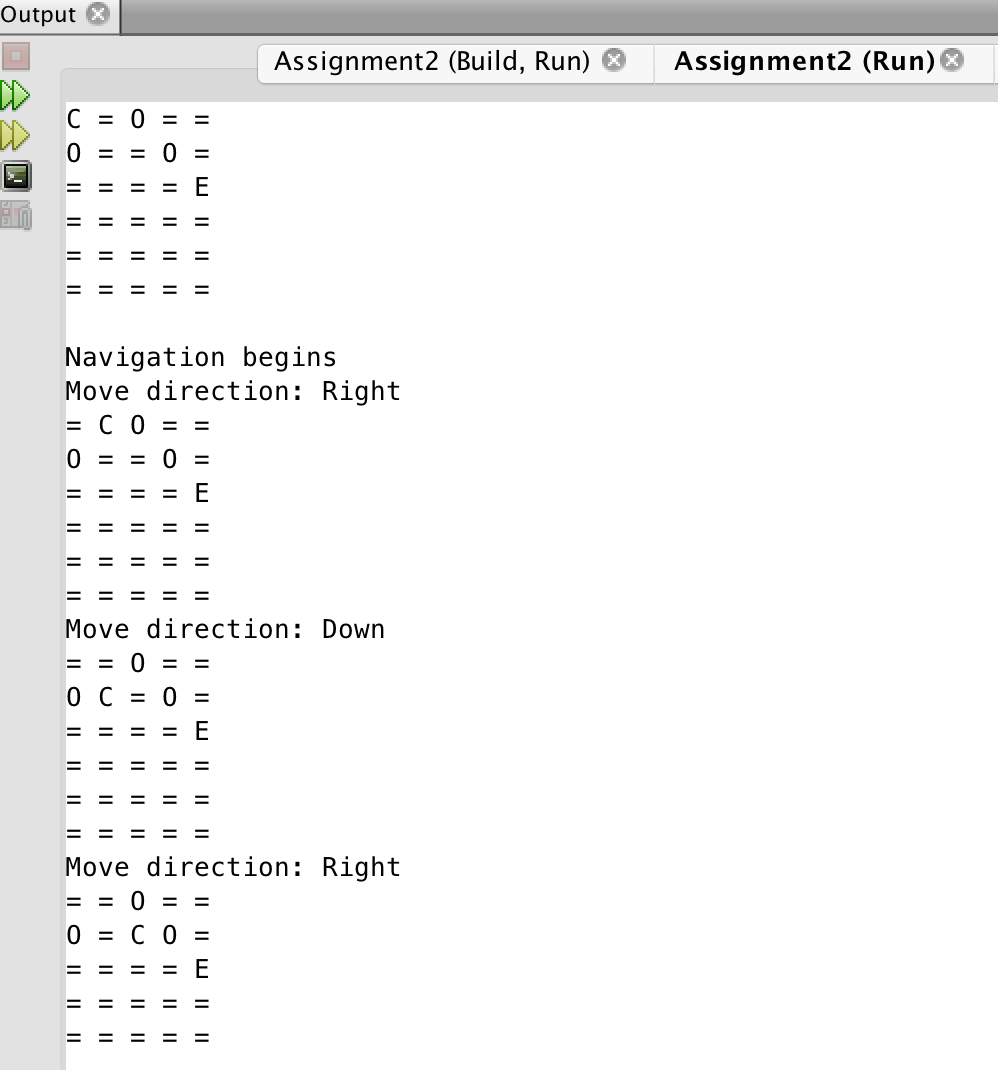
(1,1)

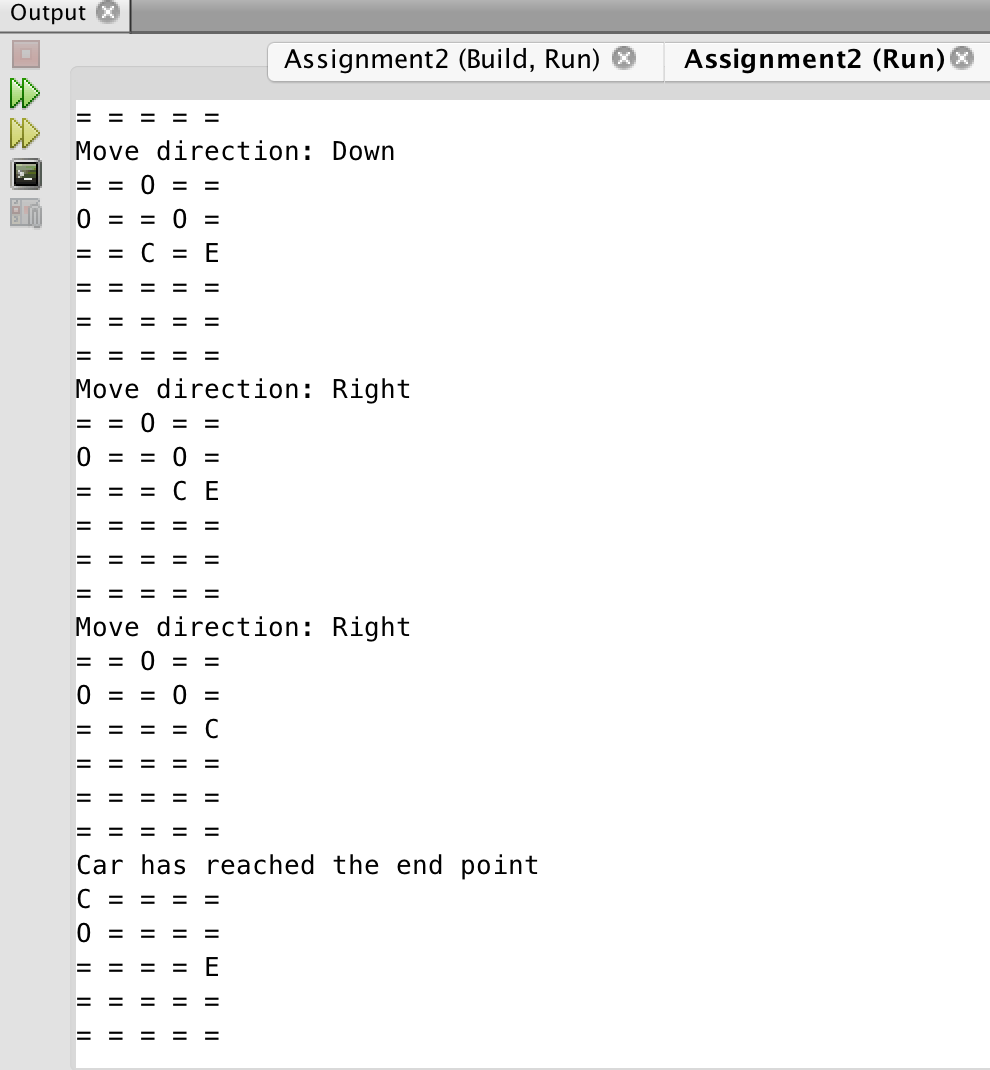
(3,5)

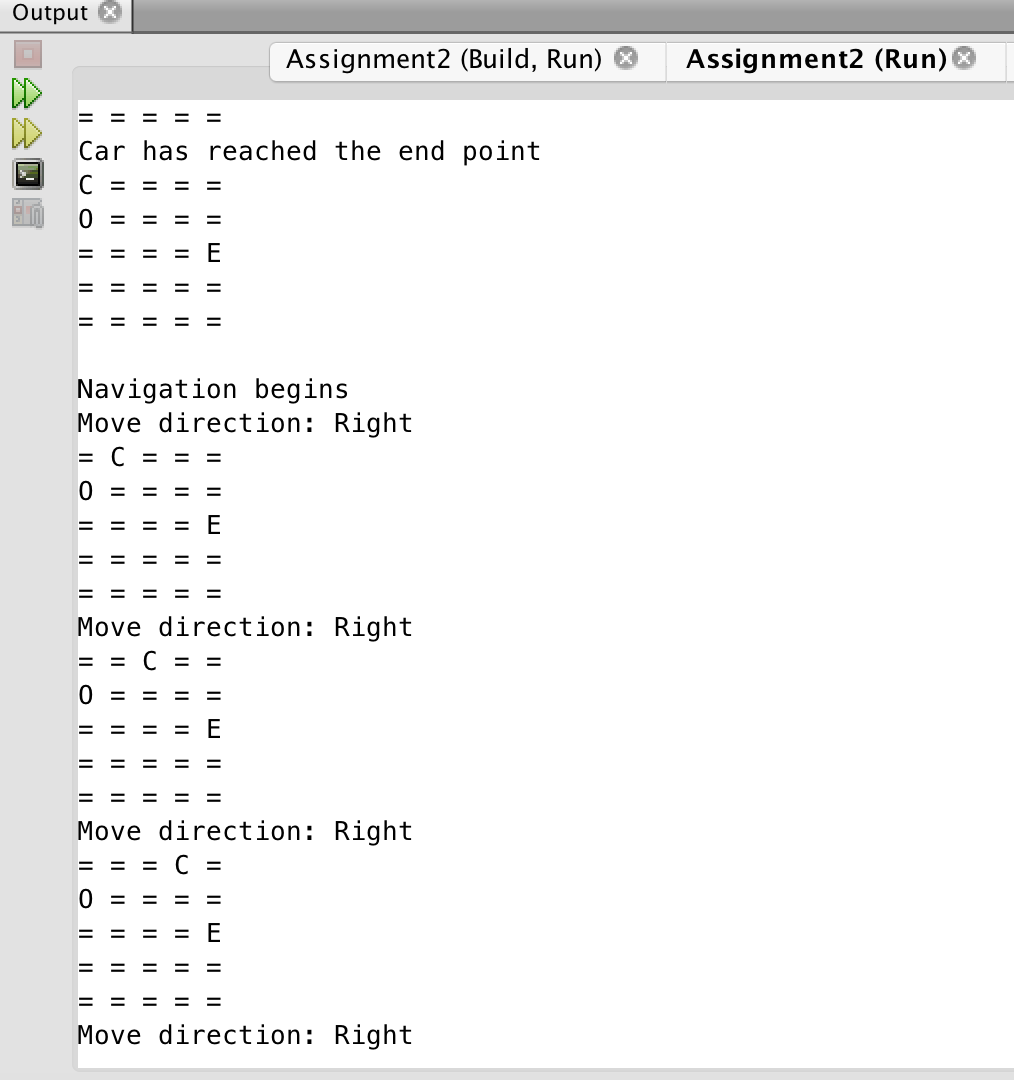
(2,1)

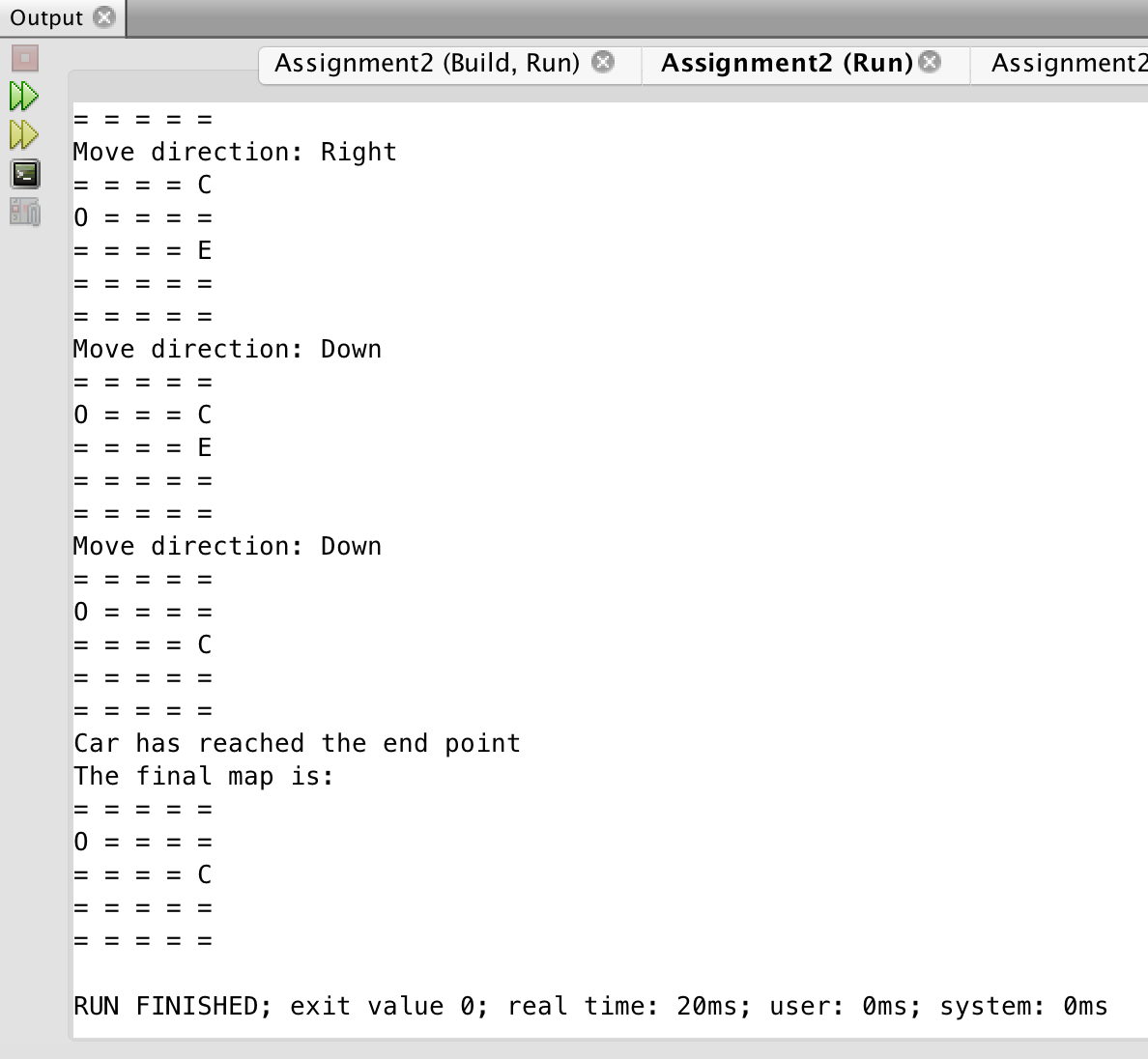
-1

**Output screenshots:**

****

****

****

****

**Things learned:**

* Concepts of operator overloading.
* Modularization of the source code.
* Implementation of the logic for the navigation purpose.
* Dynamic memory allocation.
* Friend functions.
* Multi-dimensional array and pointer concepts.

**Works Cited:**

1. <https://stackoverflow.com>
2. <http://www.tutorialspoint.com/>
3. <http://www.cplusplus.com/>
4. <http://www.geeksforgeeks.org/>
5. <https://isocpp.org/>
6. [www.codeproject.com](http://www.codeproject.com)
7. [www.sourcetricks.com](http://www.sourcetricks.com)
8. [www.dreamincode.net](http://www.dreamincode.net)