

```

Script started on 2022-09-26 08:56:13-05:00 [TERM="xterm-256color" TTY="/dev/pts/1"
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$ pwd
/home/students/pu06439/CSC122/Portfolio I/Projects/Let's MAP it out
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$ cat MapMain.info
*****
* Upanshu Parekh                                CSC122-001 *
*                                     Project: Let's MAP it out... *
*                                     *
* This program allows users to essentially play City Management *
* Simulator! Users can (1) add cities with a 20-or-less character name *
* and an x-y coordinate location, (2) calculate distances between their *
* added cities and (3) print out a list of cities that they have added. *
*                                     *
* Project Base Level: 4 *
* Options Selected: *
* Level 2: Allow user to select city lists with number or name *
* Total Project Level: 6 *
*****
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$ show-code MapMain.cpp

```

MapMain.cpp:

```

1 //Base Level: 4, Selected first option (LVL 2)
2 //Total: lvl 4+2 = 6
3 #include "cityManager.h"
4 #include <vector>
5 #include <string>
6 #include <iostream>
7 #include <climits>
8
9 using namespace std;
10 using sztp = vector<City>::size_type;
11
12 vector<City> cities{}, subCities{};
13 City newCity{}, city1{}, city2{};
14 string newName{};
15 double newX{0.0}, newY{0.0};
16 char menuSelection;
17
18 int main()
19 {
20     cout << "\nWelcome to City Management Simulator!";
21     do
22     {
23         cout << "\nWhat would you like to do?"
24             "\n\t1) E|nter city Information"
25             "\n\t2) C|alculate Distance between two cities"
26             "\n\t3) P|rint All cities"
27             "\n\t4) Q|uit"
28             "\nYour option (capitalized letter or #): ";

```

```

29     cin >> menuSelection;
30     cin.ignore(INT_MAX, '\n');
31
32     switch (menuSelection)
33     {
34         case 'E': case 'e': case '1':
35         {
36             cout << "\nYou selected [1]";
37             do
38             {
39                 cout << "\nPlease enter the name of your new city "
40                     "(Max length = " << City::MAX_CITY_NAME <<
41                     "): ";
42                 getline(cin, newName);
43             } while (newCity.set_name(newName) == false);
44
45             cout << "\nPlease enter x-coord of your new city: ";
46             cin >> newX;
47             cout << "\nPlease enter y-coord of your new city: ";
48             cin >> newY;
49             newCity.set_location(newX, newY);
50
51             cities.push_back(newCity);
52             cout << "\nSuccess! New city created: "
53                 "\"" << newCity.get_name() << "\""
54                 " located at ";
55             newCity.get_location().Output();
56             break;
57         }
58         case 'C': case 'c': case '2':
59         {
60             cout << "\nYou selected [2]";
61             if (cities.size() < 2 )
62             {
63                 cout << "\nInvalid! Need 2 or more entered cities "
64                     "for a distance calculation.";
65             }
66             else if (cities.size() == 2)
67             {
68                 cout << "\nDistance between "
69                     << cities.at(0).get_name() << " and "
70                     << cities.at(1).get_name() << " is "
71                     << cities.at(0).distance(cities.at(1))
72                     << ".";
73             }
74             else
75             {
76                 subCities = inputFindCities(
77                     "\nPlease select the first city (name or #): ",
78                     cities);
79                 if (subCities.size() == 1)
80                 {
81                     city1 = subCities.at(0);
82                 }

```

```

83     else if (subCities.size() > 1)
84     { // ^ Multiple cities of same name
85         cout << "\nMultiple cities of same name found! ";
86         city1 = inputFindCities(
87             "\nPlease select first city from the "
88             "list below with a numeric selection: ",
89             subCities, true).at(0);
90         //has to return a vector<City>
91         // with a single element
92     }
93
94     bool valid;
95     do
96     {
97         valid = true;
98         subCities = inputFindCities(
99             "\nPlease select the second city "
100             "(name or #):",
101             cities);
102         if (subCities.size() == 1)
103         {
104             city2 = subCities.at(0);
105         }
106         else if (subCities.size() > 1)
107         { // ^ Multiple cities of same name
108             cout << "\nMultiple cities of same name "
109                 "found! ";
110             city2 = inputFindCities(
111                 "\nPlease select second city from "
112                 "the list with a numeric selection: ",
113                 subCities, true).at(0);
114             //has to return a vector<City>
115             // with a single element
116         }
117         if (searchCityVector(city2, cities) ==
118             searchCityVector(city1, cities))
119         {
120             cout << "\nInvalid value! Second city must "
121                 "be different from first city. "
122                 "Try again.";
123             valid = false;
124         }
125     } while (valid == false);
126     cout << "\nDistance between \""
127         << city1.get_name()
128         << "\" and \""
129         << city2.get_name()
130         << "\" is "
131         << city1.distance(city2) << ".";
132     }
133     break;
134 }
135 case 'P': case 'p': case '3':
136 {

```

```

137         cout << "\nYou selected [3]";
138         printCityVector(cities);
139         break;
140     }
141     case 'Q': case 'q': case '4':
142     {
143         cout << "\nYou selected [4]";
144         cout << "\nThank you for using City Management "
145             "Simulator. Goodbye!";
146         break;
147     }
148     default:
149     {
150         cout << "\nInvalid selection! Try again.";
151         break;
152     }
153 }
154 } while (menuSelection != 'Q' && menuSelection != 'q' &&
155     menuSelection != '4');
156
157 return 0;
158 }

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code point.h

point.h:

```

1  /*
2  NOTE: I am aware that programs that are given to me in the project
3  description are to be used "as is". But the remaining warnings I was
4  receiving were from this "as is" class, so I had to edit it.
5  I confirmed that I could edit this with you.
6  */
7
8  #ifndef POINT_CLASS_HEADER_INCLUDED
9  #define POINT_CLASS_HEADER_INCLUDED
10
11 // A 2D point class
12 class Point
13 {
14     double x, // x coordinate of point
15         y; // y coordinate of point
16
17 public:
18
19     // CPP compiler told me to use member initialization lists for these
20     // 2 c'tors:
21     Point(void)
22         : x{0.0}, y{0.0}
23     {}
24     Point(double new_x, double new_y)
25         : x{new_x}, y{new_y}
26     {}

```

```

27
28 // = default helped fix depreacted copy warnings
29 Point(const Point & p) = default;
30
31 void Output(void) const; // output this point
32 void Input(void); // input this point
33
34 // distance between this point and other
35 double distance(const Point & other) const;
36 // point in middle of this point and other
37 Point midpoint(const Point & other) const;
38
39 double get_x(void) const { return x; } // accessors
40 double get_y(void) const { return y; }
41
42 void set_x(double new_x); // mutators
43 void set_y(double new_y);
44
45 Point flip_x(void) const; // new point is this one flipped
46 Point flip_y(void) const; // about specified axis
47
48 Point shift_x(double move_by) const; // new point is this one
49 Point shift_y(double move_by) const; // shifted move_by in the
50 // specified direction
51 };
52
53 #endif

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code point.cpp

point.cpp:

```

1  /*
2   NOTE: I am aware that programs that are given to me in the project
3   description are to be used "as is". But the remaining warnings I was
4   receiving were from this "as is" class, so I had to edit it.
5   I confirmed that I could edit this with you.
6  */
7
8  #include "point.h"
9
10 #include <iostream>
11 #include <cmath>
12
13 using namespace std;
14
15 // read standard 2D point notation (x,y) -- ignore
16 // window dressing
17 void Point::Input(void)
18 {
19     char dummy;
20     cin >> dummy >> x >> dummy >> y >> dummy;
21     return;

```

```

22 }
23
24 // output standard 2D point notation (x,y)
25 void Point::Output(void) const
26 {
27     cout << '(' << x << ", " << y << ')';
28     return;
29 }
30
31 // calculate distance between two 2D points --
32 // the one that called us and the argument
33 double Point::distance(const Point & other) const
34 {
35     return sqrt(pow(x-other.x, 2.0) +
36                 pow(other.y-y, 2.0));
37 }
38
39 // calculate midpoint between two 2D points --
40 // the one that called us and the argument
41 Point Point::midpoint(const Point & other) const
42 {
43     return Point((x+other.x)/2.0, (other.y+y)/2.0);
44 }
45
46 // set coordinates to programmer-specified values
47 void Point::set_x(double new_x)
48 {
49     x = new_x; // no error checking since anything is legal
50     return;
51 }
52
53 // set coordinates to programmer-specified values
54 void Point::set_y(double new_y)
55 {
56     y = new_y; // no error checking since anything is legal
57     return;
58 }
59
60 // in point.h the default copy c'tor was specified instead to fix
61 // a lot of deprecated operator= warnings
62
63 // construct Point as copy of previous point
64 // Point::Point(const Point & p)
65 // {
66 //     x = p.x;
67 //     y = p.y;
68 // }
69
70 // compiler told me to use member initialization lists instead here
71
72 // construct Point by default -- no values specified
73 // Point::Point(void)
74 // {
75

```

```

76 //      x = y = 0.0;
77 // }
78
79 // // construct Point given initial x,y values
80 // Point::Point(double new_x, double new_y)
81 // {
82 //     set_x(new_x);
83 //     set_y(new_y);
84 // }
85
86 // creates a point flipped about the x axis from us
87 Point Point::flip_x(void) const
88 {
89     return Point(x,-y);
90 }
91
92 // creates a point flipped about the y axis from us
93 Point Point::flip_y(void) const
94 {
95     return Point(-x,y);
96 }
97
98 // creates a point shifted along the x axis from us
99 Point Point::shift_x(double move_by) const
100 {
101     return Point(x+move_by,y);
102 }
103
104 // creates a point shifted along the y axis from us
105 Point Point::shift_y(double move_by) const
106 {
107     return Point(x,y+move_by);
108 }

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code city.h

city.h:

```

1  #pragma once
2
3  #include "point.h"
4  #include <string>
5
6  class City
7  {
8      Point location;
9      std::string name;
10
11  public:
12
13      static const std::string::size_type MAX_CITY_NAME{20};
14
15      City(void)

```

```

16      : location{},
17        name{"New City"}
18  {}
19
20  City(const std::string & newName, const Point & newLoc = Point() )
21      : City{}
22  {
23      set_name(newName);
24      set_location(newLoc);
25  }
26
27  City(const std::string & newName, const double & newX,
28        const double & newY )
29      : City{}
30  {
31      set_name(newName);
32      set_location(newX, newY);
33  }
34
35  // this was the only way I could find
36  // to prevent a "implicitly-declared operator= is deprecated"
37  // warning, have to use default copy c'tor
38  City(const City & c) = default;
39
40  //member functions
41
42  inline
43  double distance(const City & other) const
44  { return location.distance(other.location); }
45
46  inline
47  Point get_location(void) const { return location; }
48
49  void set_location(const Point & newLoc);
50
51  void set_location(const double & x, const double & y);
52
53  inline
54  std::string get_name(void) const { return name; }
55
56  bool set_name(const std::string & newName);
57  };

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code city.cpp

city.cpp:

```

1  #include "city.h"
2
3  void City::set_location(const Point & newLoc)
4  {
5      location.set_x(newLoc.get_x());
6      location.set_y(newLoc.get_y());

```

```

7 }
8
9 void City::set_location(const double & x, const double & y)
10 {
11     location.set_x(x);
12     location.set_y(y);
13 }
14
15 bool City::set_name(const std::string & newName)
16 {
17     bool okay{false};
18     if (newName.size() <= MAX_CITY_NAME)
19     {
20         name = newName;
21         okay = true;
22     }
23     return okay;
24 }

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code cityManager

cityManager.h:

```

1 #pragma once
2
3 #include "city.h"
4 #include <vector>
5 #include <string>
6
7 void printCityVector (const std::vector<City> & cityVector);
8
9 std::vector<City>::size_type
10 searchCityVector (const std::string & cityName,
11                  const std::vector<City> & cityVector,
12                  const std::vector<City>::size_type & start = 0);
13
14 std::vector<City>::size_type
15 searchCityVector (const City & city,
16                  const std::vector<City> & cityVector,
17                  const std::vector<City>::size_type & start = 0);
18
19 std::vector<City>
20 inputFindCities(const std::string & prompt,
21                const std::vector<City> & cityVector,
22                const bool & numericSel = false);

```

pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ show-code cityManager

cityManager.cpp:

```

1 #include "cityManager.h"
2 #include <vector>

```

```

3 #include <string>
4 #include <iostream>
5 #include <cstdlib>
6 #include <cmath>
7
8 using namespace std;
9 using sztp = vector<City>::size_type;
10
11 void printCityVector (const vector<City> & cityVector)
12 {
13     sztp vecSize{cityVector.size()};
14     cout << "\nCity List: ";
15     for (sztp i{0}; i != vecSize; ++i)
16     {
17         cout << "\n\t[" << i + 1 << "] \"" << cityVector.at(i).get_name()
18             << "\" located at ";
19         cityVector.at(i).get_location().Output();
20     }
21     cout << '\n';
22     return;
23 }
24
25 sztp searchCityVector (const string & cityName,
26                       const vector<City> & cityVector,
27                       const sztp & start)
28 {
29     sztp size{cityVector.size()};
30     for (sztp i = start; i != size; ++i)
31     {
32         if (cityVector.at(i).get_name() == cityName)
33         {
34             return i;
35         }
36     }
37     return -1;
38 }
39
40 sztp
41 searchCityVector (const City & city,
42                  const vector<City> & cityVector,
43                  const sztp & start)
44 {
45     string cityName{city.get_name()},
46         cVecName{};
47     double cityX{city.get_location().get_x()},
48         cityY{city.get_location().get_y()},
49         cVecX, cVecY;
50     sztp size{cityVector.size()};
51     for (sztp i = start; i != size; ++i)
52     {
53         cVecName = cityVector.at(i).get_name();
54         cVecX = cityVector.at(i).get_location().get_x();
55         cVecY = cityVector.at(i).get_location().get_y();
56     }

```

```

57     if (cVecName == cityName &&
58         fabs(cityX - cVecX) <= 1e-6 && fabs(cityY - cVecY) <= 1e-6)
59     {
60         return i;
61     }
62 }
63 return -1;
64 }
65
66 //prompts the user for city selection, finds cities in main
67 //vector that have the same name as user input
68 //detects if user wishes to input just a position from the list
69 //instead of a name
70 vector<City> inputFindCities(const string & prompt,
71                             const vector<City> & cityVector,
72                             const bool & numericSel)
73 {
74     // ^ default value = false
75     vector<City> subCities{};
76     string input{};
77     sztp cityIndex{0}, atollResult;
78     printCityVector(cityVector);
79     do
80     {
81         cout << prompt;
82         getline(cin, input);
83
84         //atoll() returns a long long int, sztp is usually unsigned
85         //long long int, both are 64 bits usually
86         //if atoll() returns negative, index will underflow, but it is
87         //an invalid position 99% of the time with the sztp datatype
88         atollResult = atoll(input.c_str());
89         cityIndex = atollResult - 1;
90
91         if (numericSel == true || atollResult != 0)
92         {
93             if (cityIndex >= cityVector.size())
94             {
95                 cout << "\nInvalid selection! Try again.";
96             }
97             else if (cityIndex < cityVector.size())
98             {
99                 subCities.push_back(cityVector.at(cityIndex));
100                 return subCities;
101             }
102         }
103     } while (cityIndex >= cityVector.size() && numericSel == true);
104     //if numeric selection requested and invalid conversion, then ask
105     //user again until it is a valid value for cityIndex (> 1)
106
107     do
108     {
109         cityIndex = 0;
110         do

```

```

111     {
112         cityIndex = searchCityVector(input, cityVector, cityIndex);
113         if (cityIndex != static_cast<sztp>(-1))
114         {
115             // ^ CPP was complaining about old-style casts
116             subCities.push_back(cityVector.at(cityIndex));
117             ++cityIndex;
118         }
119         while (cityIndex != static_cast<sztp>(-1));
120         // ^ CPP was complaining about old-style casts
121         if (subCities.size() == 0)
122         {
123             cout << "\nNo city with that name found! Try again.";
124             cout << prompt;
125             getline(cin, input);
126         }
127         while (subCities.size() == 0);
128         //if input string is valid num -> stored in cityIndex
129         return subCities;
130     }
131 }
132
133 pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$ CPP MapMain.cpp point
134 MapMain.cpp***
135 city.cpp...
136 cityManager.cpp...
137 point.cpp...

```

```

138 pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out$ ./MapMain.out
139
140 Welcome to City Management Simulator!
141 What would you like to do?
142     1) E|nter city Information
143     2) C|alculate Distance between two cities
144     3) P|rint All cities
145     4) Q|uit
146 Your option (capitalized letter or #): E
147
148 You selected [1]
149 Please enter the name of your new city (Max length = 20): CanIReallyEnterACityName!
150
151 Please enter the name of your new city (Max length = 20): Chicago
152
153 Please enter x-coord of your new city: 1.5
154
155 Please enter y-coord of your new city: 2.5
156
157 Success! New city created: "Chicago" located at (1.5, 2.5)
158 What would you like to do?
159     1) E|nter city Information
160     2) C|alculate Distance between two cities
161     3) P|rint All cities
162     4) Q|uit
163 Your option (capitalized letter or #): 2

```

You selected [2]
Invalid! Need 2 or more entered cities for a distance calculation.
What would you like to do?
1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): e

You selected [1]
Please enter the name of your new city (Max length = 20): New York

Please enter x-coord of your new city: 25.8

Please enter y-coord of your new city: 13.2

Success! New city created: "New York" located at (25.8, 13.2)
What would you like to do?
1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): C

You selected [2]
Distance between Chicago and New York is 26.5515.
What would you like to do?
1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): p

You selected [3]
City List:
[1] "Chicago" located at (1.5, 2.5)
[2] "New York" located at (25.8, 13.2)

What would you like to do?
1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): 1

You selected [1]
Please enter the name of your new city (Max length = 20): New York

Please enter x-coord of your new city: 34.2

Please enter y-coord of your new city: 37.4

Success! New city created: "New York" located at (34.2, 37.4)
What would you like to do?

1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): 1

You selected [1]
Please enter the name of your new city (Max length = 20): Atlantis

Please enter x-coord of your new city: -834.25

Please enter y-coord of your new city: 93.58

Success! New city created: "Atlantis" located at (-834.25, 93.58)
What would you like to do?
1) Enter city Information
2) Calculate Distance between two cities
3) Print All cities
4) Quit
Your option (capitalized letter or #): 2

You selected [2]
City List:
[1] "Chicago" located at (1.5, 2.5)
[2] "New York" located at (25.8, 13.2)
[3] "New York" located at (34.2, 37.4)
[4] "Atlantis" located at (-834.25, 93.58)

Please select the first city (name or #): New York

Multiple cities of same name found!
City List:
[1] "New York" located at (25.8, 13.2)
[2] "New York" located at (34.2, 37.4)

Please select first city from the list below with a numeric selection: 9

Invalid selection! Try again.
Please select first city from the list below with a numeric selection: 2

City List:
[1] "Chicago" located at (1.5, 2.5)
[2] "New York" located at (25.8, 13.2)
[3] "New York" located at (34.2, 37.4)
[4] "Atlantis" located at (-834.25, 93.58)

Please select the second city (name or #):3

Invalid value! Second city must be different from first city. Try again.
City List:
[1] "Chicago" located at (1.5, 2.5)
[2] "New York" located at (25.8, 13.2)
[3] "New York" located at (34.2, 37.4)
[4] "Atlantis" located at (-834.25, 93.58)

Please select the second city (name or #):2

Distance between "New York" and "New York" is 25.6164.
What would you like to do?

- 1) Enter city Information
- 2) Calculate Distance between two cities
- 3) Print All cities
- 4) Quit

Your option (capitalized letter or #): c

You selected [2]

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

Please select the first city (name or #): 4

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

Please select the second city (name or #):New York

Multiple cities of same name found!

City List:

- [1] "New York" located at (25.8, 13.2)
- [2] "New York" located at (34.2, 37.4)

Please select second city from the list with a numeric selection: 1

Distance between "Atlantis" and "New York" is 863.798.
What would you like to do?

- 1) Enter city Information
- 2) Calculate Distance between two cities
- 3) Print All cities
- 4) Quit

Your option (capitalized letter or #): 2

You selected [2]

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

Please select the first city (name or #): 321321

Invalid selection! Try again.

No city with that name found! Try again.

Please select the first city (name or #): 1

No city with that name found! Try again.

Please select the first city (name or #): Chicago

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

Please select the second city (name or #):4

Distance between "Chicago" and "Atlantis" is 840.698.
What would you like to do?

- 1) Enter city Information
- 2) Calculate Distance between two cities
- 3) Print All cities
- 4) Quit

Your option (capitalized letter or #): P

You selected [3]

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

What would you like to do?

- 1) Enter city Information
- 2) Calculate Distance between two cities
- 3) Print All cities
- 4) Quit

Your option (capitalized letter or #): 3

You selected [3]

City List:

- [1] "Chicago" located at (1.5, 2.5)
- [2] "New York" located at (25.8, 13.2)
- [3] "New York" located at (34.2, 37.4)
- [4] "Atlantis" located at (-834.25, 93.58)

What would you like to do?

- 1) Enter city Information
- 2) Calculate Distance between two cities
- 3) Print All cities
- 4) Quit

Your option (capitalized letter or #): Q

You selected [4]

Thank you for using City Management Simulator. Goodbye!pu06439@ares:~/CSC122/Portf
pu06439@ares:~/CSC122/Portfolio I/Projects/Let's MAP it out\$ exit
exit

Script done on 2022-09-26 09:06:19-05:00 [COMMAND_EXIT_CODE="0"]