```
1  //*************************************************************************************************
2  //                     HW 1
3  //                     Name:  Edgar Zapata
4  //                     3013 Date: Date of Submission (9/22/2021)
5  //*************************************************************************************************
6  //                   Program:
7  //                      Using the STL list libray this program will make a database
8  //                      to list records of users inputs.
9  //                      Will be  using the librays set functions to pop elements
10 //                      from the head and tail. push elements to the rear.
11 //                      Display the the datebase with the contents
12 //
13 //*************************************************************************************************
14
15 //*************************************************************************************************
16 //                  FOR ALL FUNCTIONS function Name:
17 //                       struct student:
18 //                       created the data for the linkedlist to be adding on the elements
19 //                       for the datebase
20 //
21 //*************************************************************************************************
22 //*************************************************************************************************
23 //          How to run program:
24 //              Use visual studio code or visual studio and run the code from there.
25 //              Or use a termnial and c++ or g++ the program. Then ./a.out to run
26 //
27 //*************************************************************************************************
28
29
30
31
32 #include <iostream>
33 #include <list>                                                                  // open the list library
34 #include <string>
35 using namespace std;
36                                                                                  // struct to make the elements in the linked list easier to pass
37 struct student
38 {
39
40     string first_name;                                                           // user first name
41     string last_name;                                                            // user last name
42     char gender;                                                                 // gender
```

```
43      int id;                                                         //
user id
44      int age;                                                        //
user age
45
46                                                                      //
default constructor to be able to pass the elements
47      student(string fname, string lname, char g, int student_age, int
student_id)
48      {
49          id = student_id;                                            //
seting these values to be equal to varvable list in the struct
50          first_name = fname;
51          last_name = lname;
52          gender = g;
53          age = student_age;
54      }
55  };
56
57  int main()
58  {
59
60      list<student> List_1;                                           //
creating the linked list with the stl list library
61      list<student>::iterator list1;                                  //
iterator list1 to be able to point to the elements or access list ones
elements
62
63      string fname;
64      string lname;
65      char gender;
66      int id;
67      int age;
68                                                                      //
giving the user to input the first element
69      cout << "enter the first  name" << endl;
70      cin >> fname;
71      cout << "enter the last  name" << endl;
72      cin >> lname;
73      cout << "enter the id" << endl;
74      cin >> id;
75      cout << "enter age" << endl;
76      cin >> age;
77      cout << "enter the gender" << endl;
78      cin >> gender;
79
80      List_1.push_back(student(fname, lname, gender, age, id));       //
using pushback to place the values from the rear
81
82      while (true)                                                    //
while loop to keep looping for the user to able to add more users
83      {
84          cout << "do you want to enter more. Enter Y or y for yes?" << endl;
85          char input_new;
86          cin >> input_new;
87          if (input_new == 'Y' || input_new == 'y')                   // if
statment to active the user inputs
88          {
89              cout << "enter the first  name" << endl;
90              cin >> fname;
```

```
91              cout << "enter the last  name" << endl;
92              cin >> lname;
93              cout << "enter the id" << endl;
94              cin >> id;
95              cout << "enter age" << endl;
96              cin >> age;
97              cout << "enter the gender" << endl;
98              cin >> gender;
99              List_1.push_back(student(fname, lname, gender, age, id));
100             continue;                                                   // to
keep looping the while loop untill break
101         }
102         else
103         {
104             break;                                                      //
kills the while loop
105         }
106     }
107
108     cout << endl;
109     for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        //
for loop to display the elements in  list 1
110                                                                         //
the for loop is traversing the linked list to the end
111     {
112         int id = list1->id;                                             //
this for accessing the data from the struct will be doing this for all the
data types
113         int age = list1->age;
114         char gender = list1->gender;
115         string fname = list1->first_name;
116         string lname = list1->last_name;
117
118         cout << fname << " " << lname << " - "                          //
couts out the first name, last name , and id of the users
119              << "id:" << id;
120         cout << endl;
121     }
122
123     cout << endl;
124
125     cout << "Do you want to remove the head enter Y or y for yes?" << endl;

126     char input;
127     cin >> input;
128
129     if (input == 'Y' || input == 'y')                                   // if
statment to evaulate the user input
130     {
131         if (List_1.empty())                                             //
shows the list is empty for safety purpose to not error the program
132         {
133             cout << "empty list can't pop anything" << endl;
134         }
135         else
136         {
137             List_1.pop_front();                                         //
removes the head of the linked list
138         }
139     }
```

```
140     else                                                    // if
    user does not want to remove the head
141     {
142         cout << "Head not popped" << endl;
143     }
144
145     cout << endl;
146
147     for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        //
    outputs the new list with the removed head
148     {
149         int id = list1->id;
150         int age = list1->age;
151         char gender = list1->gender;
152         string fname = list1->first_name;
153         string lname = list1->last_name;
154
155         cout << fname << " " << lname << " - "
156              << "id:" << id;
157         cout << endl;
158     }
159     cout << endl;
160
161     cout << "Do you want to remove the tail enter Y or y for yes" << endl;
162     char input_2;
163     cin >> input_2;
164     cout << endl;
165
166     if (input_2 == 'Y' || input_2 == 'y')
167     {
168
169         if (List_1.empty())                                     //
    shows the list is empty for safety purpose to not error the program
170         {
171             cout << "empty list can't pop anything" << endl;
172         }
173         else
174         {
175             List_1.pop_back();                                  //
    pops the element from the rear
176         }
177     }
178     else
179     {
180         cout << "Tail not popped" << endl;                      // if
    user choose not to pop the rear
181     }
182     cout << endl;
183
184     for (list1 = List_1.begin(); list1 != List_1.end(); list1++)        //
    displays new list with tailed being removed
185     {
186         int id = list1->id;
187         int age = list1->age;
188         char gender = list1->gender;
189         string fname = list1->first_name;
190         string lname = list1->last_name;
191
192         cout << fname << " " << lname << " - "
193              << "id:" << id;
```

```
194         cout << endl;
195     }
196 
197     cout << endl;
198 
199     while (true)                                                    //
    while to ask the user if wants to add more user
200     {
201         cout << "Do you want to enter more. Enter Y or y for yes?" << endl;
202         char input_new;
203         cin >> input_new;
204         if (input_new == 'Y' || input_new == 'y')
205         {
206             cout << "enter the first  name" << endl;
207             cin >> fname;
208             cout << "enter the last  name" << endl;
209             cin >> lname;
210             cout << "enter the id" << endl;
211             cin >> id;
212             cout << "enter age" << endl;
213             cin >> age;
214             cout << "enter the gender" << endl;
215             cin >> gender;
216             List_1.push_back(student(fname, lname, gender, age, id));
217             continue;
218         }
219         else
220         {
221             break;
222         }
223     }
224 
225     cout << endl;
226 
227                                                                     //
    display just the first name
228     for (list1 = List_1.begin(); list1 != List_1.end(); list1++)
229     {
230         int id = list1->id;
231         int age = list1->age;
232         char gender = list1->gender;
233         string fname = list1->first_name;
234         string lname = list1->last_name;
235 
236         cout << fname;
237         cout << endl;
238     }
239 
240     return 0;
241 }
```