```
1  //Bernard J. Gole Cruz, CS 202-2002, Assignment 6, problem 3
2  //This program implement link-list with exception handling
3  #include <iostream>
4  #include <stdlib.h>
5  #include <iomanip>
6
7  using namespace std;
8
9  //global variable
10 //id number starts from 100
11 int I=100;
12
13 class RecordInfo{
14
15
16 private:
17     //structure for car list
18     struct Car{
19         int id;             //value for id
20         string make;        //value for make
21         int price;          //value for price
22         int year;           //value for year
23         Car *next;          //pointer to next node
24            };
25       Car *head;  //list head pointer
26
27 public:
28     //constructor
29      RecordInfo();
30     //destructor
31      ~RecordInfo();
32
33     //link list operations
34     void insertnode(int, string, int, int);
35     void deletenode(int, string, int, int);
36     void displayList();
37 };
38
39
40 //function prototype and definitions
41
42 //constructor
43 RecordInfo::RecordInfo(){
44      //set head to null
45      head = NULL;
46 }
47
48 //destructor
49 RecordInfo::~RecordInfo(){
50     //to traverse the list
51     Car *nPtr;
52     //point to next node
53     Car *nextnode;
54
55     //set nPtr as head
56     nPtr = head;
57
58     //scan the list and delete each node
59     while (nPtr != NULL){
60         nextnode = nPtr->next;
61         //delete current node
62         delete nPtr;
63         //set nPtr to nextnode
64         nPtr = nextnode;
65     }
66 }
```

```
67  //add node at any position
68  void RecordInfo::insertnode(int ID, string MAKE, int YEAR, int PRICE){
69
70      Car *newnode;
71      //create new node
72      newnode = new Car;
73
74      int position;
75
76      //prompt user for position,keep prompting until correct input
77      cout << "Enter position: ";
78      while(!(cin >>position) ){
79              cout << "Enter position: ";
80              cin >> position;
81              cin.clear();
82              cin.ignore(100,'\n');
83                      }
84      //prompt user for make,keep prompting until correct input
85      cout << "Enter make: ";
86         while(!(cin >> MAKE) ){
87              cout << "Enter make: ";
88              cin >> MAKE;
89              cin.clear();
90              cin.ignore(100,'\n');
91                      }
92      //prompt user for price,keep prompting until correct input
93      cout << "Enter price: ";
94         while(!(cin >> PRICE) ){
95              cout << "Enter price: ";
96              cin >> PRICE;
97              cin.clear();
98              cin.ignore(100,'\n');
99                      }
100     //prompt user for year,keep prompting until correct input
101     cout << "Enter year: ";
102         while(!(cin >> YEAR) ){
103             cout << "Enter year: ";
104             cin >> YEAR;
105             cin.clear();
106             cin.ignore(100,'\n');
107                     }
108     //clear screen in each iteration
109     system("CLS");
110
111     //assign values to a node
112     newnode->id = I;
113     newnode->make = MAKE;
114     newnode->price = PRICE;
115     newnode->year = YEAR;
116
117     //set new node to null
118     newnode->next = NULL;
119
120     //insert new node in first position if list is empty
121     if (position == 1){
122         newnode->next = head;
123         head = newnode;
124         //update id starts from 100
125         I++;
126         return;
127     }
128
129     //to traverse the list
130     Car *nPtr;
131     //make new node
132     nPtr = head;
```

```
133      //insert node at any position
134      for(int i=0; i<position-2; i++){
135          nPtr = nPtr->next;
136      }
137      newnode->next = nPtr->next;
138      nPtr->next = newnode;
139      //update id starts from 100
140      I++;
141  }
142
143  //remove node from list
144  void RecordInfo::deletenode(int ID, string MAKE, int YEAR, int PRICE ){
145
146      //to traverse the list
147      Car *nPtr;
148      //point to previous node
149      Car *prev;
150
151      //delete node using id
152      cout <<"Enter id of car to be remove:"; cin >> ID;
153
154      //clear screen in each iteration
155      system("CLS");
156      nPtr->id = ID;
157
158      //do nothing if list is empty
159      if(!head)
160          return;
161
162      //check if if the first node match the item to be deleted
163      if(head->id == ID){
164          nPtr = head->next;
165          delete head;
166          head = nPtr;
167          }
168      else{
169          //set nodeptr as head
170          nPtr = head;
171
172          //skip nodes that do not match
173          while(nPtr != NULL && nPtr->id != ID){
174              prev = nPtr;
175              nPtr = nPtr->next;
176              }
177          //link previous node to next node
178          if(nPtr){
179              //delete node that has the matching item
180              prev->next = nPtr->next;
181              delete nPtr;
182              }
183          }
184  }
185
186
187  //display list
188  void RecordInfo::displayList(){
189      //scan the list
190      Car *nodePtr;
191      //set nodePtr to head
192      nodePtr = head;
193
194      cout << endl;
195      cout << endl;
196      cout << "CAR MANAGEMENT" << endl;
197      cout << "Car List:" << endl;
198
```

```
199      //traverse the list and display the elements
200      while(nodePtr){
201
202          cout << left << setw(6) << nodePtr->id << left << setw(12) << nodePtr->make
<<
203          left << setw(12) << nodePtr->price << left << setw(10) << nodePtr->year <<
endl;
204
205          nodePtr = nodePtr->next;
206      }
207       cout << "----------" << endl;
208  };
209
210  //menu
211  void menu(){
212
213      //display choice
214      cout << "Options:" << endl;
215      cout << "1. Add Car" << endl;
216      cout << "2. Remove Car" << endl;
217      cout << "5. Exit" << endl;
218
219  };
220
221  //prompt user
222  void prompt(RecordInfo &obj, int &choice)
223  {
224      int Id, Year, Price;
225      string Make;
226
227
228      //keep prompting if user choose number outside the menu
229      try{
230      menu();
231         cout << "Enter: ";
232         if(!(cin >>choice) ){
233              cin.clear();
234              cin.ignore(100,'\n');
235              throw choice;
236                      }
237          }
238         catch(int choice){
239              throw;
240              }
241
242      int value;
243      //choices in menu
244      switch (choice){
245      case 1:
246          //insert node in list
247          obj.insertnode(Id, Make, Year, Price);
248           //display list contents
249          obj.displayList();
250          break;
251
252      case 2:
253          //remove node from list
254          obj.deletenode(Id, Make, Year, Price);
255          //display list contents
256          obj.displayList();
257
258          break;
259
260      case 5:
261          //exit menu
262              exit(0);
```

```
263
264         default:
265             //keep prompting until a correct choice is made
266             cout << "try again!!" << endl;
267         }
268 };
269
270
271 int main(){
272     int select;
273     //RecordInfo object, create car list
274     RecordInfo list;
275     bool success = false;
276
277
278     cout << endl;
279     cout << endl;
280     cout << "CAR MANAGEMENT" << endl;
281     cout << "Car List:" << endl;
282     cout << "----------";// <<;// endl;
283     cout << endl;
284
285     //exception handling
286     while(true){
287         try{
288             prompt(list, select);
289             success = true;
290             }
291         catch(...){
292             }
293         }
294
295
296     return 0;
297 }
```