```
 1 #ifndef _STUDENT
 2 #define _STUDENT
 3
 4 #include <iostream>
 5 #include <string>
 6 #include <algorithm>
 7 using namespace std;
 8
 9 /**
10  * Class Student
11 */
12 class Student {
13 private:
14     int IDNumber;
15     string name;
16     string major;
17     int year;
18
19 public:
20     Student(int id = 0, string name = "", string major
    = "", int year = 0);
21     // getters and setters
22     int getIDNumber();              // returns IDNumber
23     void setIDNumber(int year);     // sets the
   IDNumber
24     string getName();               // returns Name
25     void setName(string name);      // sets the name
26     string getMajor();              // returns Major
27     void setMajor(string major);    // sets the major
28     int getYear();                  // returns Year
29     void setYear(int year);         // set the year
30
31     // Overload as friend functions
32     // Comparison should be based on the IDNumber only
    assuming each
33     // student has a unique ID. No need to compare the
    other fields.
34     friend bool operator== (const Student& lhs, const
```

```
34 Student& rhs);
35     friend bool operator>  (const Student& lhs, const
   Student& rhs);
36     friend bool operator<  (const Student& lhs, const
   Student& rhs);
37 };
38
39 #endif
40
```

```cpp
/*****************************************************
*********************
* Homework 6
* author          :Ajinkya Joshi
* Date            :April 5, 2023
* File name       :main.cpp
* Purpose         :Contains the main menu options to
add, remove, etc.
 *****************************************************
*********************/

// All libraries and header files used in the program
#include <iostream>
#include "Student.h"
#include "BinarySearchTree.h"
#include <fstream>
#include <iomanip>
using namespace std;

/**
 * prints the student information
 * @param stu
 * @return void
 */
void print(Student& stu)
{
    cout << setw(5) << stu.getIDNumber();
    cout << setw(10) << stu.getName();
    cout << setw(10) << stu.getMajor();
    cout << setw(10) << stu.getYear() << endl;
}

/**
 * main
 * main function for the program
 */
int main()
{
```

```
36      BinarySearchTree<Student>* stuTreePtr = new
   BinarySearchTree<Student>(); // create a new tree
37
38      // Student Variables defined only once
39      int id;
40      string name;
41      string major;
42      int year;
43      int option; // menu option
44
45      // Display Menu
46      cout<< "Menu Options:" << endl;
47      cout<< "+--------------------------+"<< endl;
48      cout<< "| [0] DISPLAY MENU" << endl;
49      cout<< "| [1] NUMBER OF RECORDS" << endl;
50      cout<< "| [2] ADD RECORD" << endl;
51      cout<< "| [3] PRINT RECORDS" << endl;
52      cout<< "| [4] RETRIEVE BY IDNO" << endl;
53      cout<< "| [5] REMOVE RECORD" << endl;
54      cout<< "| [6] REMOVE ALL RECORDS" << endl;
55      cout<< "| [7] READ RECORDS FROM FILE" << endl;
56      cout<< "| [8] TERMINATE PROGRAM" << endl;
57      cout<< "+--------------------------+" << endl;
58
59      // Menu
60      while(option != 8)
61      {
62          cout<< "===> Enter an integer menu option[0-8
   ]:";
63          cin >> option;
64
65          // Option displays menu again by looping
66          if (option == 0){
67              // Display Menu
68              cout<< "Menu Options:" << endl;
69              cout<< "+--------------------------+"<<
   endl;
70              cout<< "| [0] DISPLAY MENU" << endl;
```

```
 71                cout<< "| [1] NUMBER OF RECORDS" << endl;
 72                cout<< "| [2] ADD RECORD" << endl;
 73                cout<< "| [3] PRINT RECORDS" << endl;
 74                cout<< "| [4] RETRIEVE BY IDNO" << endl;
 75                cout<< "| [5] REMOVE RECORD" << endl;
 76                cout<< "| [6] REMOVE ALL RECORDS" << endl
    ;
 77                cout<< "| [7] READ RECORDS FROM FILE" <<
    endl;
 78                cout<< "| [8] TERMINATE PROGRAM" << endl;
 79                cout<< "+-------------------------+" <<
    endl;
 80            }
 81
 82            // Prints the number of record in the tree
 83            else if(option == 1)
 84            {
 85                if (stuTreePtr->isEmpty())
 86                    cout<< "The List is empty. No records
     found!" << endl;
 87                else
 88                    cout<< "The List contains " <<
    stuTreePtr->getNumberOfNodes() << " records" << endl;
 89            }
 90
 91            // Adds a new record to the tree
 92            else if(option == 2)
 93            {
 94                cout<< "ID: ";
 95                cin >> id;
 96                cout<< "Name: ";
 97                cin >> name;
 98                cout<< "Major: ";
 99                cin >> major;
100                cout<< "Year: ";
101                cin >> year;
102                Student* stuPtr = new Student(id, name,
    major, year);
```

```
103             stuTreePtr->add(*stuPtr);
104             cout << "One record has been added to the
     list." << endl;
105         }
106
107         // Prints all the records in the tree
108         else if(option == 3)
109         {
110             if (stuTreePtr->isEmpty())
111             {
112                 cout << "List is empty. Nothing to
    print..." << endl;
113             }
114             else
115             {
116                 cout<< "  ID#      Name     Major
         Year" << endl;
117                 stuTreePtr->inorderTraverse(print);
118                 cout<< endl;
119             }
120
121         }
122
123         // Retrieves a record from the tree
124         else if(option == 4)
125         {
126             if(stuTreePtr->isEmpty())
127             {
128                 cout << "Unable to perform this
    action on an empty list." << endl;
129             }
130             else
131             {
132                 int idno;
133                 cout << "Enter student's ID for
    record to search: ";
134                 cin >> idno;
135                 try {
```

```
136                     Student *stuPtr2 = new Student(
    idno);
137                     stuTreePtr->getEntry(*stuPtr2);
138                     Student stuPtr3 = stuTreePtr->
    getEntry(*stuPtr2);
139                     cout << "  ID#      Name
    Major      Year" << endl;
140                     print(stuPtr3);
141                 }
142                 catch (NotFoundException) {
143                     cout << "Student not found in
    records...!" << endl;
144                 }
145                 catch (PrecondViolatedExcep) {
146                     cout << "Student not found in
    records...!" << endl;
147                 }
148             }
149
150         }
151
152         // Removes a record from the tree
153         else if(option == 5)
154         {
155             if (stuTreePtr->isEmpty())
156             {
157                 cout << "Unable to perform this
    action on an empty list." << endl;
158             }
159             else
160             {
161                 int idno2;
162                 bool remove;
163                 cout<< "Enter ID Number: ";
164                 cin >> idno2;
165                 Student* stuPtr3 = new Student(idno2
    );
166                 stuTreePtr->remove(*stuPtr3);
```

```
167
168                 try {
169                     Student* stuPtr3 = new Student(
    idno2);
170                     stuTreePtr->remove(*stuPtr3);
171                 }
172                 catch(NotFoundException)
173                 {
174                     cout << "Student not found in
    records...!" << endl;
175                 }
176                 catch(PrecondViolatedExcep)
177                 {
178                     cout << "Student not found in
    records...!" << endl;
179                 }
180             }
181
182         }
183
184         // Removes all records from the tree
185         else if(option == 6)
186         {
187             if (stuTreePtr->isEmpty())
188             {
189                 cout << "Unable to perform this
    action on an empty list." << endl;
190             }
191             else
192             {
193                 cout << "All records have been
    deleted from the list." << endl;
194                 stuTreePtr->clear();
195             }
196         }
197
198         // Reads records from a file
199         else if(option == 7)
```

```
200         {
201             string fileName = "StudentRecords.txt";
202             ifstream inFile(fileName.c_str());
203             if (!inFile)
204             {
205                 cout << "File could not be opened."
     << endl;
206             }
207             else
208             {
209                 cout << "Reading records from file
    ...." << endl;
210                 while(getline(inFile, fileName))
211                 {
212                     inFile >> id;
213                     inFile >> name;
214                     inFile >> major;
215                     inFile >> year;
216                     Student* stuPtr = new Student(id
    , name, major, year);
217                     stuTreePtr->add(*stuPtr);
218                 }
219                 inFile.close();
220             }
221
222         }
223
224         // Terminates the program
225         else if(option == 8)
226         {
227             cout<< "Processing completed. Terminating
    ........" << endl;
228             break;
229         }
230
231         // Invalid option
232         else
233         {
```

```
234                 cout<< "Invalid option. Please try again
    ." << endl;
235             }
236
237         }
238
239
240         return 0;
241 }
242
```

```cpp
/*****************************************************************************
* Homework 6
* author          :Ajinkya Joshi
* Date            :April 5, 2023
* File name       :Student.cpp
* Purpose         :Contains the method implementations for the Student class.
 *****************************************************************************/

#include "Student.h"

// Constructor for the Student class that initializes the member variables with the given parameters.
Student::Student(int id, string name, string major, int year) {
    IDNumber = id;
    this->name = name;
    this->major = major;
    this->year = year;
}

// Getter function that returns the ID number of the student.
int Student::getIDNumber() {
    return IDNumber;
}

// Setter function that sets the ID number of the student.
void Student::setIDNumber(int id) {
    IDNumber = id;
}

// Getter function that returns the name of the student.
string Student::getName() {
```

```
31      return name;
32  }
33
34  // Setter function that sets the name of the student.
35  void Student::setName(string name) {
36      this->name = name;
37  }
38
39  // Getter function that returns the major of the
    student.
40  string Student::getMajor() {
41      return major;
42  }
43
44  // Setter function that sets the major of the student.
45  void Student::setMajor(string major) {
46      this->major = major;
47  }
48
49  // Getter function that returns the current year of
    the student.
50  int Student::getYear() {
51      return year;
52  }
53
54  // Setter function that sets the current year of the
    student.
55  void Student::setYear(int year) {
56      this->year = year;
57  }
58
59  // Comparison operator function that checks if two
    student objects are equal based on their ID numbers.
60  bool operator == (const Student& lhs, const Student&
    rhs) {
61      return lhs.IDNumber == rhs.IDNumber;
62  }
63
```

```
64 // Comparison operator function that checks if the ID
    number of the first student object is greater than
   the second.
65 bool operator > (const Student& lhs, const Student&
   rhs) {
66     return lhs.IDNumber > rhs.IDNumber;
67 }
68
69 // Comparison operator function that checks if the ID
    number of the first student object is less than the
   second.
70 bool operator < (const Student& lhs, const Student&
   rhs) {
71     return lhs.IDNumber < rhs.IDNumber;
72 }
73
```

```
Last login: Thu Apr  6 03:46:53 on ttys003
ajinkyajoshi@Ajinkyas-MacBook-Pro EECE2560_Homework-6 % make
g++ main.cpp Student.cpp -o output
ajinkyajoshi@Ajinkyas-MacBook-Pro EECE2560_Homework-6 % ./output
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:1
The List is empty. No records found!
===> Enter an integer menu option[0-8]:3
List is empty. Nothing to print...
===> Enter an integer menu option[0-8]:4
Unable to perform this action on an empty list.
===> Enter an integer menu option[0-8]:5
Unable to perform this action on an empty list.
===> Enter an integer menu option[0-8]:6
Unable to perform this action on an empty list.
===> Enter an integer menu option[0-8]:0
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:2
ID: 1550
Name: HENRY
Major: ECE
Year: 3
One record has been added to the list.
===> Enter an integer menu option[0-8]:2
ID: 2350
Name: MARK
Major: CS
```

```
Year: 5
One record has been added to the list.
===> Enter an integer menu option[0-8]:1
The List contains 2 records
===> Enter an integer menu option[0-8]:3
  ID#      Name     Major      Year
 1550     HENRY       ECE         3
 2350      MARK        CS         5

===> Enter an integer menu option[0-8]:4
Enter student's ID for record to search: 6500
Student not found in records...!
===> Enter an integer menu option[0-8]:4
Enter student's ID for record to search: 1550
  ID#      Name     Major      Year
 1550     HENRY       ECE         3
===> Enter an integer menu option[0-8]:5
Enter ID Number: 7500
===> Enter an integer menu option[0-8]:5
Enter ID Number: 1550
===> Enter an integer menu option[0-8]:3
  ID#      Name     Major      Year
 2350      MARK        CS         5

===> Enter an integer menu option[0-8]:0
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:6
All records have been deleted from the list.
===> Enter an integer menu option[0-8]:1
The List is empty. No records found!
===> Enter an integer menu option[0-8]:0
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
```

```
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:7
Enter File Name: StudentRecords.txt
Reading records from file....
===> Enter an integer menu option[0-8]:1
The List contains 19 records
===> Enter an integer menu option[0-8]:3
  ID#      Name     Major      Year
 1100      MARY       ECE         3
 1200      JOHN       ECE         5
 1300      ALEX        CS         4
 1700       BEN       ECE         1
 1800      JOAN        CS         2
 2100       DAN       PHY         4
 2400       KIM        CE         2
 2600      ANNE       MIE         3
 3500      NICK       PHY         4
 4100      MARK       ECE         3
 4300      JANE       CEE         1
 4400       JOY       ECE         2
 4600       TOM       PHY         1
 4700      LUKE       MIE         1
 5000      JUDE       PHY         5
 5300      LUCY        EE         2
 5400      BILL       ENG         3
 5500       EVE        CS         3
 5900       TIM       PHY         3

===> Enter an integer menu option[0-8]:0
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:5
Enter ID Number: 8500
===> Enter an integer menu option[0-8]:5
Enter ID Number: 5500
```

```
===> Enter an integer menu option[0-8]:5
Enter ID Number: 5900
===> Enter an integer menu option[0-8]:4
Enter student's ID for record to search: 5500
Student not found in records...!
===> Enter an integer menu option[0-8]:4
Enter student's ID for record to search: 1200
  ID#      Name     Major      Year
 1200      JOHN       ECE         5
===> Enter an integer menu option[0-8]:1
The List contains 17 records
===> Enter an integer menu option[0-8]:2
ID: 5950
Name: ZACK
Major: CE
Year: 3
One record has been added to the list.
===> Enter an integer menu option[0-8]:2
ID: 1000
Name: ADAM
Major: EE
Year: 2
One record has been added to the list.
===> Enter an integer menu option[0-8]:1
The List contains 19 records
===> Enter an integer menu option[0-8]:0
Menu Options:
+-------------------------+
| [0] DISPLAY MENU
| [1] NUMBER OF RECORDS
| [2] ADD RECORD
| [3] PRINT RECORDS
| [4] RETRIEVE BY IDNO
| [5] REMOVE RECORD
| [6] REMOVE ALL RECORDS
| [7] READ RECORDS FROM FILE
| [8] TERMINATE PROGRAM
+-------------------------+
===> Enter an integer menu option[0-8]:3
  ID#      Name     Major      Year
 1000      ADAM        EE         2
 1100      MARY       ECE         3
 1200      JOHN       ECE         5
 1300      ALEX        CS         4
 1700       BEN       ECE         1
 1800      JOAN        CS         2
 2100       DAN       PHY         4
 2400       KIM        CE         2
 2600      ANNE       MIE         3
```

```
 3500      NICK       PHY         4
 4100      MARK       ECE         3
 4300      JANE       CEE         1
 4400       JOY       ECE         2
 4600       TOM       PHY         1
 4700      LUKE       MIE         1
 5000      JUDE       PHY         5
 5300      LUCY        EE         2
 5400      BILL       ENG         3
 5950      ZACK        CE         3

===> Enter an integer menu option[0-8]:1
The List contains 19 records
===> Enter an integer menu option[0-8]:6
All records have been deleted from the list.
===> Enter an integer menu option[0-8]:1
The List is empty. No records found!
===> Enter an integer menu option[0-8]:3
List is empty. Nothing to print...
===> Enter an integer menu option[0-8]:7
Enter File Name: StudentRecords.txt
Reading records from file....
===> Enter an integer menu option[0-8]:3
  ID#      Name     Major      Year
 1100      MARY       ECE         3
 1200      JOHN       ECE         5
 1300      ALEX        CS         4
 1700       BEN       ECE         1
 1800      JOAN        CS         2
 2100       DAN       PHY         4
 2400       KIM        CE         2
 2600      ANNE       MIE         3
 3500      NICK       PHY         4
 4100      MARK       ECE         3
 4300      JANE       CEE         1
 4400       JOY       ECE         2
 4600       TOM       PHY         1
 4700      LUKE       MIE         1
 5000      JUDE       PHY         5
 5300      LUCY        EE         2
 5400      BILL       ENG         3
 5500       EVE        CS         3
 5900       TIM       PHY         3

===> Enter an integer menu option[0-8]:8
Processing completed. Terminating........
ajinkyajoshi@Ajinkyas-MacBook-Pro EECE2560_Homework-6 %
```