```cpp
//Grant dawson COSC 220 Project 1
#include"CourseDB.h"

using namespace std;
//defual constructor
CourseDB::CourseDB(){
  head=nullptr;
}
//deconstructor
CourseDB::~CourseDB(){
  courseNode* cursor=head;
  while(head){
    head=head->next;
    delete cursor;
    cursor=head;
  }
}
//Copy constructor
CourseDB::CourseDB(const CourseDB& old){
  if(old.head==nullptr){
    head=nullptr;
  }else{
    courseNode* oldCursor=old.head;
    while(oldCursor){
      append(oldCursor->course);
      oldCursor=oldCursor->next;
    }
  }
}
//= operator
CourseDB* CourseDB::operator=(const CourseDB& x){
  courseNode* oldCursor=x.head;
  while(oldCursor){
    append(oldCursor->course);
    oldCursor=oldCursor->next;
  }
    return (this);
}
//appends a new Course node ot the end with the values of temp
void CourseDB::append(Course temp){
  courseNode* cursor=head;
  courseNode* newNode = new courseNode;
  newNode->course=temp;
  newNode->next=nullptr;
  if(head==nullptr){
    head=newNode;
    return;
  }
  while(cursor->next){
    cursor=cursor->next;
  }
  cursor->next=newNode;
}

//update a given Course to given c
void CourseDB::update(Course c,Course newC){
  courseNode* cursor=head;
  while(cursor){
    if(cursor->course==c){
      cursor->course = newC;
    }
    cursor=cursor->next;
```

```
 63   }
 64 }
 65
 66 //removes a course that is equal too c
 67 void CourseDB::remove(Course c){
 68   courseNode* cursor=head;
 69   courseNode* prev=head;
 70
 71   while(cursor->next){
 72     prev=cursor;
 73     cursor=cursor->next;
 74     if(head->course==c){
 75       cursor=head;
 76       head=head->next;
 77       delete cursor;
 78     }else if(cursor->course==c){
 79       prev->next=cursor->next;
 80       delete cursor;
 81     }
 82   }
 83   if(cursor==head && prev==head){
 84     head=head->next;
 85     delete cursor;
 86     return;
 87   }
 88 }
 89
 90 //prints all the courses in the list
 91 void CourseDB::printAll(){
 92   if(!head){
 93     cout<<"Student not in any courses!"<<endl;
 94     return;
 95   }
 96   courseNode* cursor=head;
 97   while(cursor){
 98     //cout<<"Name: "<<cursor->course.getName()<<" Department: "<<cursor-
        >course.getDepartment()<<" Semester: "<<cursor->course.getSemester()<<" Grade:
        "<<endl;
 99     cout<<cursor->course.getDepartment()<<":"<<cursor->course.getName()<<" | Semester:
        "<<cursor->course.getSemester()<<" | Grade: "<<cursor->course.getGrade()<<endl;
100     cursor=cursor->next;
101   }
102 }
103
104
105 void CourseDB::printAllList(){
106   if(!head){
107     cout<<"Student not in any courses!"<<endl;
108     return;
109   }
110   int i=1;
111   courseNode* cursor=head;
112   while(cursor){
113     //cout<<"Name: "<<cursor->course.getName()<<" Department: "<<cursor-
        >course.getDepartment()<<" Semester: "<<cursor->course.getSemester()<<" Grade:
        "<<endl;
114     cout<<i<<")"<<cursor->course.getDepartment()<<":"<<cursor->course.getName()<<" |
        Semester: "<<cursor->course.getSemester()<<" | Grade: "<<cursor->course.getGrade()<<"
        "<<endl;
115     cursor=cursor->next;
116     i++;
117   }
```

```
118 }
119
120 //returns a course the user choose and it found by what index it was at
121 Course CourseDB::chooseCourse(int index){
122   courseNode* cursor=head;
123   for(int i=0;i<index;i++){
124     cursor=cursor->next;
125   }
126   return cursor->course;
127 }
128
129 //This returns the amount of courses in the list starting from 0
130 int CourseDB::length(){
131   courseNode* cursor=head;
132   int counter=0;
133   if(!head)
134     return counter;
135   while(cursor){
136     counter++;
137     cursor=cursor->next;
138   }
139   return counter;
140 }
141
```