

```
1  /*
2  Name: Rifa Safeer Shah
3  Class: CECS 282
4  Instructor: Minhthong Nguyen
5  */
6
7  #include <iostream>
8  #include "Course.h"
9  #include "Time.h"
10 #include "Student.h"
11
12 using namespace std;
13
14 extern int const CAPACITY = 35; //set maximum capacity of enrollment in a course
15
16 /* Default Constructor */
17 Course::Course() {
18     courseName = ""; //name of the course (CECS282)
19     courseNumber = ""; //four-digit course number (2856)
20     semester = ""; //the semester of the course - FALL, WINTER, SPRING, SUMMER ↗
21     (FALL)
22     last_date_to_enroll = Time(0); //the time of last date to enroll in seconds
23     numOfEnrolled = 0; //number of students enrolled
24     this->students = new Student[CAPACITY]; //pointer that points to array of ↗
25     students
26 } //ends Default Constructor
27
28 /* Overload Constructor */
29 Course::Course(string name, string num, string sem, Time last_date, Student* ↗
30 stdts, int numOfEnroll) {
31     this->courseName = name; //name of the course
32     this->courseNumber = num; //number of the course
33     this->semester = sem; //semester of the course
34     this->last_date_to_enroll = last_date; //last date to enroll into the course
35     this->numOfEnrolled = numOfEnroll; //number of students enrolled
36     if (numOfEnroll > CAPACITY) {
37         students = new Student[numOfEnroll];
38     }
39     this->students = new Student[CAPACITY];
40     for (int i = 0; i < numOfEnroll; i++) {
41         students[i] = stdts[i];
42     }
43 } //ends Overload Constructor
44
45 /* Copy Constructor */
46 Course::Course(const Course& c) {
47     (*this).courseName = c.courseName;
48     (*this).courseNumber = c.courseNumber;
49     (*this).semester = c.semester;
```

```
47     (*this).last_date_to_enroll = c.last_date_to_enroll;
48     (*this).numOfEnrolled = c.numOfEnrolled;
49     students = new Student[CAPACITY];
50     for (int i = 0; i < numOfEnrolled; i++) {
51         students[i] = c.students[i];
52     }
53 } //ends Copy Constructor
54
55 /* Assignment Operator */
56 Course& Course::operator=(const Course& c) {
57     (*this).courseNumber = c.courseNumber;
58     (*this).courseName = c.courseName;
59     (*this).semester = c.semester;
60     (*this).last_date_to_enroll = c.last_date_to_enroll;
61     (*this).numOfEnrolled = c.numOfEnrolled;
62     for (int i = 0; i < numOfEnrolled; i++) {
63         students[i] = c.students[i];
64     }
65     return (*this);
66 } //ends Assignment Operator
67
68 /* Destructor */
69 Course::~~Course() {
70     delete[] students;
71 } //ends Destructor
72
73
74 /* Get Student */
75 Student* Course::getStudent() const {
76     return students;
77 } //ends Get Student
78
79
80 /* Get Course Number */
81 string Course::getCourseNumber() const {
82     return courseNumber;
83 } //ends Get Course Number
84
85
86 /* Get Course Name */
87 string Course::getCourseName() const {
88     return courseName;
89 } //ends Get Course Name
90
91 /* Get Semester */
92 string Course::getSemester() const {
93     return semester;
94 } //ends Get Semester
95
```

```
96  /* Get Year */
97  string Course::getYear() const { //return the year of the semester (assume to be
    the year of the last date to enroll)
98      return last_date_to_enroll.getYear();
99  } //ends Get Year
100
101  /* Get Number of Enrollment */
102  int Course::getNumberOfEnrollment() const {
103      return numOfEnrolled;
104  } //ends Get Number of Enrollment
105
106  /* Get Last Date to Enroll */
107  string Course::getLastDateToEnroll() const { // return as Www Mmm dd hh:mm:ss
    yyyy
108      return last_date_to_enroll.toString();
109  } //ends Last Date to Enroll
110
111  /* Get Time Last Date to Enroll */
112  Time Course::getTimeLastDateToEnroll() const { // return last_date_to_enroll in
    secondsvoid
113      return last_date_to_enroll;
114  } //ends Get Time Last Date to Enroll
115
116  /* Set Number of Enrollment */
117  void Course::setNumberOfEnrollment(int ne) {
118      numOfEnrolled = ne;
119  } //ends Set Number of Enrollment
120
121  /* Set Roster */
122  void Course::setRoster(Student* stdts) {
123      Student* temp = new Student[CAPACITY];
124      for (int i = 0; i < numOfEnrolled; i++) {
125          temp[i] = stdts[i];
126      }
127      this->students = temp;
128
129  } //ends Set Roster
```