Class **Course**:

- Instance Attributes:
  - **courseName**: name of the course
  - **price**: price of the course

- Methods:
  - **Course(courseName:String)**: constructor
  - **caluclatePrice():** return the price of the course:
  - **OnlineCourse:** calculates the price by multiplying each Lecture by 50 SR
  - **InPersonCourse:** If the course is over 20 hours you pay 35SR for an hour , if it's under 20 hours you pay 50SR for an hour

  - **equals(obj:Object):** compares two courses if not equivalent return false otherwise return true
  - **setCourseName(courseName:String):** set Course name
  - **getPrice():** return price
  - **toString():** return a string representation of the course in the following format:
  - **Course: *courseName* Price: *price***

Class **OnlineCourse:**

- Instance Attributes:
  - **lecturesNum:** number of lectures

- Methods:
  - **OnlineCourse(courseName:String , lecturesNum:int):** constructor
  - **OnlineCourse(c:OnlineCourse):** copy constructor
  - **setLecturesNum(lecturesNum:int):** set the number of lectures
  - **toString():** return a string representation of the course in the following format**: type: [Online Course] ,Number of Lectures: *lectureNum***

Class **InPersonCourse:**

- Instance Attributes:
  - hours: number of hours of a course
  - date: starting date of course

- Methods:
  - **InPersonCourse(courseName:String , hours:int , date:String):** constructor
  - **InPersonCourse(c:InPersonCourse):** copy constructor
  - **setHours(hours:int):** set number hours
  - **setDate(date:String):** set starting date
  - **toString():** return a string representation of the course in the following format: **type: [In Person Course] , starting Date (*date*) ,Hours: *hours***

Class **Teacher:**

- Instance Attributes:
  - **teacherName:** teacher's name
  - **email:** teacher's email
- Class Attributes:
  - **availableCourses:** array of courses
  - **cCount:** number of courses in available courses

- Methods
  - **Teacher(teacherName:String , email:String):** constructor
  - **Teacher(size:int):** Constructor with size parameter and automatically fills information and skips register.
  - **addCourse(c:Course):** if the array is full or the Course c is already in the array return false , if Course c is instance of Online course add it as OnlineCourse , if Course c is instance of in person course add it as InPersonCourse then return true

    - deleteCourses(choice:int): if parametrized integer is out of array boundaries return false , otherwise delete Course and return true
    - search(c:course): Searches for a given course (`c`) in the student's courses, Iterates through the `availableCourses` array and compares each course with the given course (`c`). Returns the index of the course if found, otherwise returns `-1`.
    - viewAvailableCourses(): show all the available courses in the array
    - setTeacherName(teacherName:String): : updates the value of teacher's name
    - setEmail(email:String): : updates the value of teacher's email
    - toString(): returns a string representation of the contact in the following format:
    - **Teacher: *teacherName* , Email: *email***

Class **Student:**

- Instance Attributes:
    - **studentName:** name of the student
    - **age:** age of the student
    - **major:** major of the student
    - **studentCourses:** array of Course objects
    - **cCount:** counts how many courses are there

- Methods:
    - **Student(studentName:String , age:int , major:String, size: int):** Constructor
    - **Student(size : int):** Constructor with size and automatically fills information and skips register
    - **joinCourse(choice:int):** if parametrized integer is out of array boundaries or there is no space in the array return false , otherwise add the course based on choice (index) of available courses array
    - **leaveCourse(choice:int):** if parametrized integer is out of array boundaries return false , otherwise delete the course based on choice (index) of available courses array

- **search(c: Course):** Searches for a given course (`c`) in the student's courses, Iterates through the `studentCourses` array and compares each course with the given course (`c`). Returns the index of the course if found, otherwise returns `-1`.
- **viewStudentCourses():** displays student courses
- **viewAvailableCourses():**displays available courses
- **setStudentName(String: studentName):** setter for student name
- **setAge(int: age):** setter for student age
- **setMajor(String: major):** setter for student major
- **totalPrice():** total price of all the courses that the student added
- **toString():** returns a string representation of the student in the following format**: Student: *studentName* ,Age: *age* ,Major: *major***