Key concepts:

* variable/function declaration
* types (number, string, array, object, null)
* this keyword
* new keyword
* OOP js basics based on prototype
* conditional statements
* &&, ||, !== operators
* typeof operator
* console.log, console.error
* string concatenation
* array iteration
* array prototype methods

Implement the following 2 classes: “Course” and “Student”. The classes should have the following features (note: classes should be implemented using constructor functions; class/constructor keywords are to be avoided for our first implementation):

Course:

* required properties: id (number), name (string), assignedTeacher (string), studentList (Array[Students])
* id should be unique for each course
* when creating a new course, you have to validate it; if one property is missing or if a property type is other than the requested one or if the id is not unique, the course will not be created
* If the course validation fails, we should print one of the following errors in the console: “Required property is missing”, “Invalid property type” or “Course with id <courseId> already exists”.
* If the course validation succeed, “Course was successfully created” message should be printed in the console
* you should have the possibility to add or delete students; the delete operation will be made based on the student’s id
* after adding a student to a course, “Student with id <studentId> was successfully added to this course ” message should be print in the console
* when deleting a student, if the id is invalid we should print the following message in the console: “Student id is invalid resulting in failing to delete the specified student from this course”
* If deleting the student was successful, “Student with id <studentId> was successfully deleted” message should be print in the console (e.g: “Student with id 5 was successfully deleted from this course”)
* Course should contain a method to print all its students in the console

Student:

* required properties: id (number), firstName (string), lastName (string), gender (string)
* additional properties: address (string), hobbies (string)
* id should be unique for each student
* when creating a new student, you have to validate it; if one **required** property is missing or if a property type is other than the requested one or if the id is not unique, the student will not be created
* If the student validation fails, we should print one of the following errors in the console: “Required property is missing”, “Invalid required property type”, “Invalid additional property type”, “Student with id <studentId> already exists”
* If an **additional** property is not set when creating the student, the value of that property will be set to null
* If the student validation succeed, “Student was successfully created” message should be printed in the console

Input/output data:

var historyCourse = new Course(1, 'history', 'Victoria Cobbett', []); *//should output 'Course was successfully created'*

var frenchCourse = new Course('3', 'french', 'Josselin Bourseiller', []); *//should output 'Invalid property type'*

var englishCourse = new Course(2, 'english', 'Andrea Barrett', []); *//should output 'Course was successfully created'*

var mathematicsCourse = new Course(2, 'mathematics', 'Jack Connor', []); *//should output 'Course with id 2 already exists'*

var physicsCourse = new Course(10); *//should output 'Required property is missing'*

var martinStudent = new Student(1, 'Martin', 'Lawrence', 'M'); *//should output 'Student was successfully created'*

var kellyStudent = new Student ('3', 'Freddy', 'Kelly', 'M', '3497 James Avenue') *//should output 'Invalid required property type'*

var kellyStudent = new Student (2, 'Freddy', 'Kelly', 'M', '3497 James Avenue') *//should output 'Student was successfully created'*

var maxStudent = new Student(2, 'Max', 'Austin', 'M', '4026 Lee Avenue', 'sports') *//should output 'Student with id 2 already exists'*

var maxStudent = new Student(3, 'Max', 'Austin', 'M', '4026 Lee Avenue', 'sports') *//should output 'Student was successfully created'*

var aliyaCollins = new Student(4, 'Aliya', 'Collins', 'F', '353 Oliverio Drive', 19) *//should output 'Invalid additional property type'*

var aliyaCollins = new Student(4, 'Aliya', 'Collins', 'F', '353 Oliverio Drive', 'cooking, reading') *//should output 'Student was successfully created'*

var norahCollins = new Student(5, 'Norah') *//should output 'Required property is missing'*

console.log(martinStudent.hobbies) *//should output null*

historyCourse.addStudent(martinStudent); *//should output 'Student with id 1 was successfully added to this course'*

historyCourse.addStudent(kellyStudent); *//should output 'Student with id 2 was successfully added to this course'*

englishCourse.addStudent(martinStudent); *//should output 'Student with id 1 was successfully added to this course'*

historyCourse.printStudentList();

*//should output*

*// [*

*// Student {*

*// id: 1,*

*// firstName: 'Martin',*

*// lastName: 'Lawrence',*

*// gender: 'M',*

*// address: null,*

*// hobbies: null*

*// },*

*// Student {*

*// id: 2,*

*// firstName: 'Freddy',*

*// lastName: 'Kelly',*

*// gender: 'M',*

*// address: '3497 James Avenue',*

*// hobbies: null*

*// }*

*// ]*

englishCourse.printStudentList();

*//should output*

*// [*

*// Student {*

*// id: 1,*

*// firstName: 'Martin',*

*// lastName: 'Lawrence',*

*// gender: 'M',*

*// address: null,*

*// hobbies: null*

*// }*

*// ]*

historyCourse.deleteStudent(101); *//should output 'Student id is invalid resulting in failing to delete the specified student from this course'*

historyCourse.deleteStudent(1); *//should output 'Student with id 1 was successfully deleted from this course'*

historyCourse.printStudentList();

*//should output*

*// [*

*// Student {*

*// id: 2,*

*// firstName: 'Freddy',*

*// lastName: 'Kelly',*

*// gender: 'M',*

*// address: '3497 James Avenue',*

*// hobbies: null*

*// }*

*// ]*