Utility

+ Utility()

+ initProjectPool(filename : string, projectPool[] : Project) : void

+ initStudentPool(filename : string, studentPool[] : Project) : void

+ initClassSectionPool(filename : string, classSectionPool[] : ClassSection) : void

+ initProjectStudentSkills(filename : string, projectPool[] : Project) : void

+ getSizeOfJson(filename : string, key : string) : int

+ getProjectXskill(projectPool[] : Project, i : int, j : int) : int

+ getSkillXstudent(studentPool[] : Student, i : int, j: int) : int

+ calcProjectXStudentMatrix(students : vector<Student>, projects : vector<Project>) : vector<vector<int>>

+ projectTypePartition(projectPool[] : Project, numProjects : int, t0 : char, t1 : char, t2 : char) : void

+ projectPriorityPartition(projectPool[] : Project, numProjects : int, t0 : int, t1 : int, t2 : int) : void

+ PriorityPartition(projectPool[] : Project, numProjects : int, t0 : int, t1 : int, t2 : int) : void

+ classSectionTypePartition(classSectionPool[] : ClassSection, numClassSections : int, t0 : char, t1 : char) : void

+ printIntMatrix(a : vector<vector<int>>) : void

+ ProjectToSectionPercentages(studentList : vector<vector<Student>>, projectList : vector<Project>, numProjects : int, NumOfClassSections : int) : int**

+ arrayProjectToSectionPercentages(projectPool[] : Project, studentPool[] : Student, classSectionPool[] : ClassSection, percentMatrix[] : int, numProjects : int, numStudents : int, numClassSections : int, numSkills : int) : void

+ projectToSectionAssignment(projectPool[] : Project, studentPool[] : Student, classSectionPool[] : ClassSection, numProjects : int, numStudents : int, numClassSections : int, numSkills : int, studentsInSections[] : int) : void

+ makeProjectJSON(numProj : int, numSkill : int) : void

+ makeProjectCSV(numProj : int, numSkill : int) : void

+ makeStudentJSON(numStud : int, numSkill : int, studentsFromCanvas : vector<vector<Student>>) : void

+ makeStudentCSV(numStud : int, numSkill : int) : void

+ calc_projects(numStudents : int, teamSize : int, minTeamSize : int) : int

+ NumOfTeamsOf4(int numStudents, int teamSize) : int

+ toCSVsse(string filename) : vector<vector<string>>

+ toCSVcse(string filename) : vector<vector<string>>

+ csvToProjectsVector(filename : string, projectPool[] : Project, numProjects : int) : vector<Project>

+ getQuizID(quizName : string, filename : string) : int

+ getAssignmentID(quiz_ID : int, filename : string) : int

+ getCategoryID(courseID : int, filename : string) : int

+ getGroupID(course_ID : int, filename : string) : int

+ getSurveyAnswers(students : vector <Student>, assignment_ID : int, filename : string) : vector<Student>

+ getStudentsFromJson(filename : string) : vector<Student>

+ ~Utility()

ClassSection

+ OfficialClassID:int
+ ClassID:int
+ Course_Name:string
+ Course_Code:string
+ Type:char
+ Enrollment:int

+ operation1(params):returnType
+ ClassSection()
+ ClassSection(cID:int, t:char)

Learn about this template

UML class diagrams map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects.

To customize this template:

- *Click on any shape and type the information you would like to include.*
- *Add and arrange class shapes as needed.*
- *Update cardinality.*
  - *Click on a line and navigate to the properties bar to adjust the endpoints.*
  - *Click on a line and hover over the gear icon to add multiplicities.*
  - *Add additional lines by hovering over a shape and clicking the red dot*

UML Class Diagram Tutorials

(Hold Shift + ⌘ or Ctrl, then click)

Watch a UML class diagram tutorial

Read about UML class diagrams

Watch Lucidchart basic tutorials