MainApp

Dependency line <<create>> (*For CourseRegistrationMarkEntry class)

```
CourseRegistrationMarkEntry app = new CourseRegistrationMarkEntry();
```

CourseRegistrationMarkEntry

Composition line (*For Student, Staff, Module and Course classes)

```
private ArrayList<Student> students;
private ArrayList<Staff> staffs;
private ArrayList<Module> modules;
private ArrayList<Course> courses;
```

Dependency line <<create>> (*For Student, Staff, Module and Course classes)

```
students = new ArrayList<Student>();
staffs = new ArrayList<Staff>();
modules = new ArrayList<Module>();
courses = new ArrayList<Course>();
```

Dependency line <<create>> (count as 1 line) (*For Group class)

```
course.setLecture(new Group(moduleId + "-" + year + "-" + semester + "-LT", staffId,
venue, vacancy));
course.getTutorials().add(new Group(moduleId + "-" + year + "-" + semester + "-TUT"
+ count, staffId, venue, vacancy));
course.getLaboratories().add(new Group(moduleId + "-" + year + "-" + semester + "-
LAB" + count, staffId, venue, vacancy));
```

Dependency line<<create>> (count as 1 line) (*For StudentCourse class)

```
searchForStudentById(studentId).getRegisteredCourses().add(new
StudentCourse(course.getCourseId()));
```

Dependency line<<create<> (*For Weightage class)

```
course.setWeightage(new Weightage(examWeightage, courseworkWeightage));
```

Dependency line (count as 1 line) (*For Module class)

```
protected void addCourse()
{
   Module module;
.
.
.
}
private Module searchForModuleById(String moduleId)
```

Dependency line (count as 1 line) (*For Course class)

```
protected void registerStudentToCourse()
{
.
.
.
Course course;
.
.
.
}
```

```
private Course searchForCourseByIdYearSem(String moduleId, int year, int semester)
private Course searchForCourseById(String courseId)
```

Dependency line (count as 1 line) (*For Group class)

```
private Group searchForGroupById(String groupId, ArrayList<Group> groupList)
protected void registerStudentToCourse()
{
.
.
.
Group selectedLecture = null;
Group selectedTutorial = null;
Group selectedLaboratory = null;
.
.
.
}
```

Dependency line(count as 1 line) (*For StudentCourse class)

```
protected void enterStudentCourseworkMark()
{
.
.
.
StudentCourse studentCourse;
.
.
.
}
protected void enterStudentExamMark()

.
.
.
StudentCourse studentCourse;
.
.
.
}
```

Dependency line(count as 1 line) (*For Student class)

```
protected void enterStudentCourseworkMark()
{
.
.
.
Student student;
.
.
.
}
```

Dependency line(count as 1 line) (*For Staff class)

```
private Staff searchForStaffById(String staffId)
```

Course

Composition line 1 to 1 (*For Group and Weightage class)

```
private Group lecture;
private Weightage weightage;
```

Composition line 1 to many (Count as 1 line) (*For Group class)

```
private ArrayList<Group> tutorials;
private ArrayList<Group> laboratories;
```

Dependency line <<create>> (*For Weightage class)

```
weightage = new Weightage(0, 0);
```

Dependency line (count as 1 line) (*For Weightage class)

```
public Weightage getWeightage()
public void setWeightage(Weightage weightage)
```

Dependency line (count as 1 line) (*For Group class)

```
public void setLecture(Group lecture)
public Group getLecture()
public void setTutorials(ArrayList<Group> tutorials)
public ArrayList<Group> getLaboratories()
public void setLaboratories(ArrayList<Group> laboratories)
```

Generalisation (Inheritance) (*For Module class)

```
public class Course extends Module
```

Student

Composition line 1 to many (*For StudentCourse class)

```
private ArrayList<StudentCourse> registeredCourses;
```

Dependency line (count as 1 line) (*For StudentCourse class)

```
public StudentCourse getRegisteredCourseById(String courseId)
public ArrayList<StudentCourse> getRegisteredCourses()
```

Generalisation (Inheritance) (*For Person class)

```
public class Student extends Person
```

Staff

Generalisation (Inheritance) (*For Person class)

```
public class Staff extends Person
```