CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Computing@CIT

LINEAR DATA STRUCTURES AND ALGORITHMS

Lab03: Object Oriented Programming.

BACKGROUND.

The folder **/src** contains the following files:

- **myMain.java:** This class tests the functionality of the other 2 classes.
- **student.java:** This class models a student and its functionality.
- **module.java:** This class models a module and its functionality.

The folder **/doc** contains the documentation of the project. In particular:

- **myMain.html:** Contains the description of the class myMain.java.
- **student.html:** Contains the description of the class student.java.
- **module.html:** Contains the description of the class module.java.

Note: Try to solve the lab without looking at student.html and module.html.
Consult them only if you got stuck with the attributes and functions you are required to model.

EXERCISE.

Implement the student.java and module.java classes following the description of next pages.

1. MODEL A STUDENT.

We need to model a student. In our case, a student is characterised by:

- **A name.** Example: “Jack”, “Mary”, etc.
- **An age.** Example: 18, 25, etc.
- **Is it a first year’ student.** Example: Yes or no.

Besides that, we want a student to do some functionality on its birthday. When this happens:

- Its age gets incremented.
- The message “Happy birthday *Name*, we hope you are enjoying your *(first)* year in college” is printed by the screen.
  Note: *name* is the name of the student, and the word *first* only appears for first year students.

EXERCISE.

Complete the Java class student.java to model the concept of a student.

The class must include:

1) The student attributes (with what characterises a student).
2) A constructor (to create a new student object).
3) Get and set methods (so as to access and update the attributes of the object).
4) An extra function, modelling the aforementioned functionality for a student birthday.
   public void birthday();

2. MODEL A COLLEGE MODULE.

We need to model a college module. In our case, a module is characterised by:

- **A name.** Example: "Linear Data Structures and Algorithms", "Object Oriented Programming", etc.
- **A code.** Example: 1234, 7428, etc.
- **Number of students.** Example: 0, 17, 46, etc.
- **Student Registered.** Example: No students or [("Jack", 20, first year), ("Mary", 25, not first year), ("Peter", 18, first year)]
- **A maximum number of students.** Example: 30, 50, etc.

Besides that, we want a module to be able to do the following functionality:

I) Register a new student, which will be placed after all students previously registered.
II) Get the information of the i-est student registered to the module.

EXERCISE.

Complete the Java class module.java to model the concept of a module.

The class must include:

1) The module attributes (with what characterises a module).
2) A constructor (to create a new module object).
   Note: When creating a new module, we just assume it contains 0 students.
3) Get and set methods (so as to access and update the attributes of the object).
4) Two extra functions, modelling the aforementioned functionality for student registration and student information retrieval.
   public student getStudentInfo(int i);
   public void registerStudent(student s);