

CSIT213 Autumn Session 2024

Assignment 1

Published on 28 February 2024 **Due: Saturday 23 March, 11:30 pm**

Total Marks: 15 marks

Scope

This assignment is related to the UML classes diagrams, Java classes' definitions and implementations, arrays, Strings and data input/output.

Please read very carefully the information listed below.

General Java coding requirements:

- Create your programs with good programming style and form using proper blank spaces, indentation, and braces to make your code easy to read and understand.
- Create identifiers with sensible names.
- Add proper comments to describe your code segments where they are necessary for readers to understand what your code intends to achieve.
- Logical structures and statements are properly used for specific purposes.
- Read the assignment specification carefully, and make sure that you follow the direction in this assignment. In **every assignment source file** that you will submit on this subject, you must put the following information in the header of your program:

```
/*-----  
    My name:  
    My student number:  
    My course code: CSIT213  
    My email address:  
    Assignment number: 1  
-----*/
```

A submission procedure is explained at the end of the specification.

It is recommended to solve the problems before attending the laboratory classes in order to efficiently use supervised laboratory time.

A submission marked by Moodle as `Late` is treated as a late submission no matter how many seconds it is late.

A policy regarding late submissions is included in the subject outline.

A submission of compressed files (zipped, gzipped, rared, tared, 7-zipped, lhzed, ... etc) is **not allowed**. The compressed files will not be evaluated.

An implementation that does not compile due to one or more syntactical or processing errors scores no marks. It is expected that all tasks included in **Assignment 1** will be solved **individually without any cooperation** from the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for the assessment task.

Tasks

In this assignment, you are required to design and implement a system for Assignment, Student and Subject in Java. This system helps to manage students enrolled in subjects and related assignments.

Implementation

Your program shall contain at least the following classes for the system.

Define a class `Subject` in a source file `Subject.java` that contains private data fields:

- `code`: Subject code. It is a String type.
- `title`: Subject title. It is a String type.
- `credits`: Credits a subject worth. It is an integer type.
- `school`: School name that offers the subject It is a String type.

Implement Java methods in the file `Subject.java` that include:

- A parameterized constructor that assigns values to all data fields.
- A public method `getCode()` that returns the subject code.
- A public method `toString()` that returns the string value of all private data fields. See the examples of the processing for the details of the format of the `toString` method.

Define a class `Student` in a source file `Student.java` that contains private data fields:

- `number`: Student number. It is an integer type.
- `name`: Student name. It is a String type.
- `dob`: Date of birth. It is a String type.
- `degree`: Student degree. It is a String type.
- `codes`: Enrolled subject codes. It is a String array type. Assume the maximum number of enrolled subjects is 10.
- `cnt`: Number of enrolled subjects. It is an integer type.

Implement Java methods in the file `Student.java` that include:

- A parameterized constructor that assigns values to all data fields. Create a String array with the maximum number of elements 10 for codes. Initially, there is no subject enrolled.
- A public method `getNumber()` that returns the student number.
- A public method `addCode(String code)` that adds a new subject code in the String array `codes`.
- A public method `toString()` that returns the String value of all private data fields. See the examples of the processing for the details of the format of the `toString()` method.

Define a class `Assignment` in a source file `Assignment.java` that contains private data fields:

- `code`: Subject code. It is a String type.
- `number`: Assignment number. It is an integer type.
- `dueDate`: Assignment due date. It is a String type.
- `totalWorth`: Assignment marks that total worth. It is an integer type.

Implement Java methods in the file `Assignment.java` that include:

- A parameterized constructor that assigns values to all data fields.
- A public method `getCode()` that returns the subject code.
- A public method `getNumber()` that returns the assignment number.

- Public method `toString()` that returns the `String` value of all private data fields. See the examples of the processing for the details of the format of the `toString` method.

Define a class `mainManagement` in a source file `mainManagement.java` that contains private data members:

- `MAXNUM`: A constant integer. Set the constant value to 100.
- `students`: An array of students. It is a `Student` array type.
- `cntStudents`: Number of students in the array `students`. It is an integer type.
- `subjects`: An array of subjects. It is a `Subject` array type.
- `cntSubjects`: Number of subjects in the array `subjects`. It is an integer type.
- `assignments`: An array of assignments. It is an `Assignment` array type.
- `cntAssignments`: Number of students in the array `students`. It is an integer type.

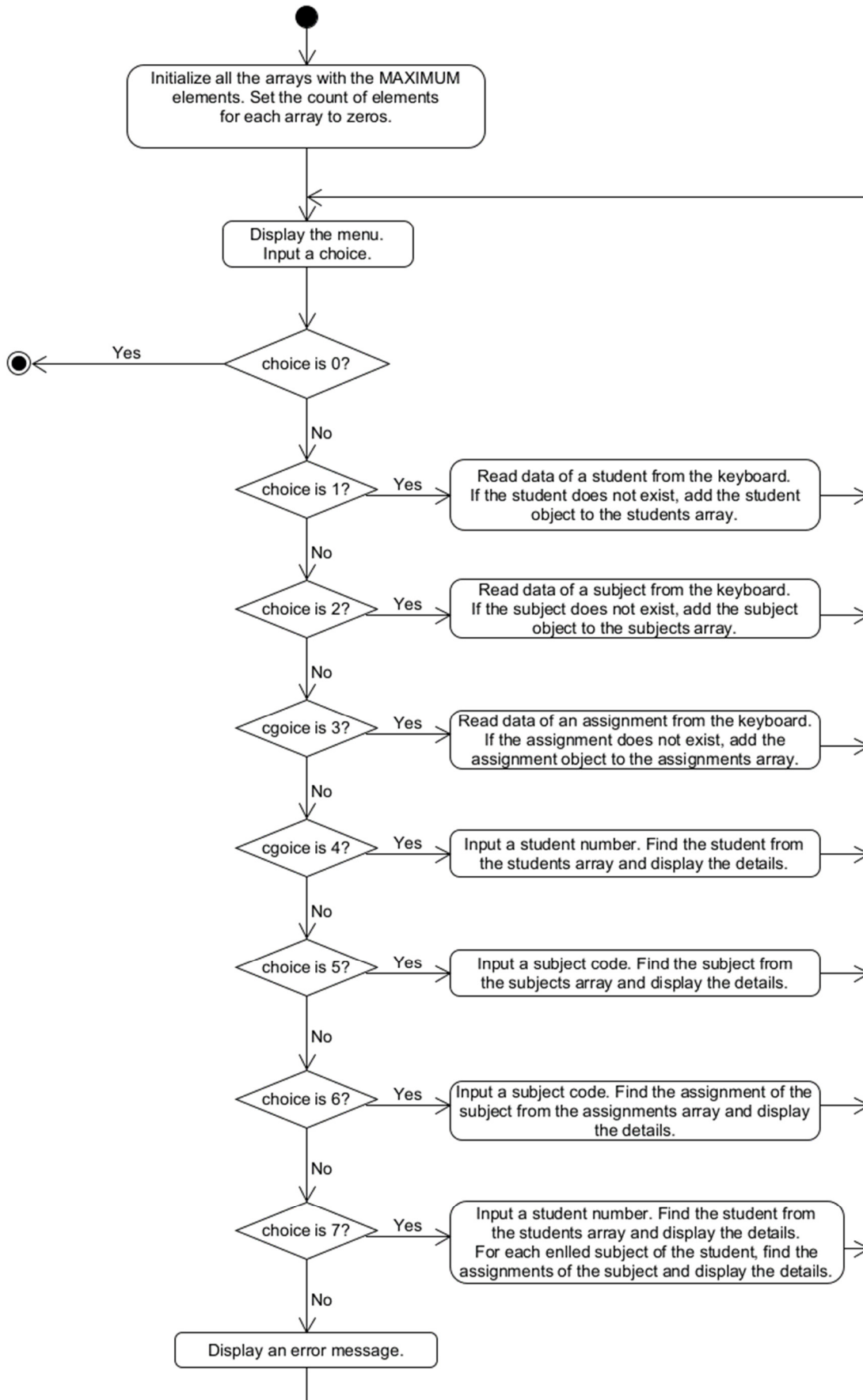
Implement Java methods in the file `mainManagement.java` that include:

- Default constructor that initializes values to all data fields.
- Public static method `main(String[] args)` that starts the program, displays a menu, get input choices from a user, and then process required activities. The UML activity diagrams are shown below.

See the processing examples for the details.

- Other necessary methods for all the activities. It is recommended that implement at least one method for each activity.

UML activity diagram



Compilation and testing

Compile your program by using the javac command.

```
javac mainManagement.java
```

Process your program by using the java command.

```
java mainManagement
```

Test your program for all the activities. See the examples of the processing results below for more details.

Processing example

Examples of application processing are given below. The user's inputs are highlighted in red colour.

```
java mainManagement
```

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 1

Student number: 12345

Student name: Alice Malk

Date of birth: 1/2/2000

Degree: BSc

Enrolled subjects (Separated by whitespace): CSIT110 CSIT111 CSIT115
CSIT213 CSCI235

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 1

Student number: 22345

Student name: Bob Kart

Date of birth: 12/3/2001

Degree: BIT

Enrolled subjects (Separated by whitespace): CSIT110 CSIT115 CSIT213

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 1

Student number: 12345

The student 12345 exists. Cannot add a student

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 1

Student number: 32345

Student name: Carl Newman

Date of birth: 11/2/1999

Degree: BSc

Enrolled subjects (Separated by whitespace): CSIT111 CSIT213 CSCI235

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 2

Subject code: CSIT110

Subject title: Python

Credits: 6

Offered by: School of CIT

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 2

Subject code: CSIT213

Subject title: Java Programming

Credits: 6

Offered by: School of CIT

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 2

Subject code: CSIT110

The subject CSIT110 exists. Cannot add a subject.

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 2

Subject code: CSIT115

Subject title: Database Designs and Security

Credits: 6

Offered by: School of CIT

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 3

Subject code: CSIT115

Assignment number: 1

Due date: 10/8/2023

Total worth: 10

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 3

Subject code: CSIT115

Assignment number: 2

Due date: 10/9/2023

Total worth: 10

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 3

Subject code: CSIT115

Assignment number: 1

The assignment 1 of the subject CSIT115 exists. Cannot add an assignment.

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 3

Subject code: CSIT213

Assignment number: 1

Due date: 19/8/2023

Total worth: 15

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 3

Subject code: CSIT110

Assignment number: 1

Due date: 10/8/2023

Total worth: 10

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 4

Student number: 22345

Student number: 22345

Name: Bob Kart

Date of birth: 12/3/2001

Degree: BIT

Subjects: CSIT110 CSIT115 CSIT213

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 4

Student number: 54321

Student 54321 does not exist

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 5

Subject code: CSIT213

Subject code: CSIT213

Title: Java Programming

Credits: 6

Offered by: School of CIT

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 5

Subject code: CSCI235

Subject CSCI235 does not exist

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 6

Subject code: CSCI235

No assignment for the subject CSCI235

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 6

Subject code: CSIT110

Subject code: CSIT110

Assignment number: 1

Due date: 10/8/2023

Total worth: 10

1. Add a student

2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 6

Subject code: CSIT213

Subject code: CSIT213

Assignment number: 1

Due date: 19/8/2023

Total worth: 15

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 7

Student number: 12345

Student number: 12345

Name: Alice Malk

Date of birth: 1/2/2000

Degree: BSc

Subjects: CSIT110 CSIT111 CSIT115 CSIT213 CSCI235

Subject code: CSIT110

Assignment number: 1

Due date: 10/8/2023

Total worth: 10

Subject code: CSIT115

Assignment number: 1

Due date: 10/8/2023

Total worth: 10

Subject code: CSIT115

Assignment number: 2

Due date: 10/9/2023

Total worth: 10

Subject code: CSIT213

Assignment number: 1

Due date: 19/8/2023

Total worth: 15

1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit

Input a choice: 7
Student number: 32345
Student number: 32345
Name: Carl Newman
Date of birth: 11/2/1999
Degree: BsC
Subjects: CSIT111 CSIT213 CSCI235
Subject code: CSIT213
Assignment number: 1
Due date: 19/8/2023
Total worth: 15
1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit
Input a choice: 10
Incorrect choice
1. Add a student
2. Add a subject
3. Add an assignment
4. Find a student
5. Find a subject
6. Find all assignments of a subject
7. Find all assignments of a student
0. Quit
Input a choice: 0
Bye!

Deliverables

(1) UML class diagram (2 marks): Use the UMLet application tool to draw the class diagram. The class diagram shall

- contains at least the classes mentioned above;
- contains the class name, fields, and methods definitions for each class;
- use correct and sufficient UML notations;
- specify the associations between classes;
- specify the multiplicities for both sides of the associations.

Remember to use the CSIT213 palette!

Use the option File->Export as... to export a class diagram into a file in BMP format. Do not delete an exported file. You will use it as one of the solutions for your task.

Insert the BMP files into a Word file `assignment1Report.docx`.

(2) Implementation (11 marks): Implement the application according to the UML class diagrams and the UML activity diagrams described above. The program shall

- be consistent with the UML class diagrams.
- follow the conventions for naming all classes, variables, and methods.
- provide sufficient comments.
- use proper blank spaces, indentation, and braces to make your code easy to read and understand.
- follow the specified implementation steps.
- be able to repeat the main menu until the user exits the system.

(3) Compilation and test (2 marks): Compilation and test your Java program by using the command line interface.

- Please carefully compile your program. Make sure your program can pass the compilation by using the `javac` command.
- Test your program by using the `java` command.
Test your program for all the activities. See the examples of the processing results above for more details.
- **Please do not define the package in your program (a special alert for students who use IDE to complete the assignment).**

Copy and paste the compilation and testing results into the Word file `assignment1Report.docx`. When ready convert the Word file `assignment1Report.docx` into a pdf file `assignment1Report.pdf`.

Submission

Note that you have only one submission. So, make absolutely sure that you submit the correct files with the correct contents and correct types. No other submission is possible!

Submit the files **mainManagement.java**, **Assignment.java**, **Student.java**, **Subject.java**, and **assignment1Report.pdf** through Moodle in the following way:

- (1) Access Moodle at **<http://moodle.uowplatform.edu.au/>**
- (2) To login use a **Login** link located in the right upper corner of the Web page or in the middle of the bottom of the Web page
- (3) When logged select a site **CSIT213 (S223) Java Programming**
- (4) Scroll down to a section **Assignments and Submissions**
- (5) Click on a link **Assignment 1 submission**
- (6) Click on the button **Add Submission**
- (7) Upload a file **mainManagement.java** into an area **You can drag and drop files here to add them**. You can also use a link **Add...**
- (8) Repeat step (7) for the files **Assignment.java**, **Student.java**, **Subject.java** and **assignment1Report.pdf**.
- (9) Click on the checkbox with a text attached: **By checking this box, I confirm that this submission is my own work, ...** in order to confirm the authorship of your submission
- (10) Click on a button **Save changes**

End of specification