School of Computing and Information Technology
University of Wollongong

Session: **Autumn 2023**
Lecturer: **Tianbing Xia**

CSIT1121 Object Oriented Design and Programming
Assignment 1

Published on 27 February 2023

Scope

This assignment is related to the class conceptual schema, Java classes' definition and implementation.

Please read very carefully the information listed below.
This assignment contributes to 10% of the total evaluation in the subject CSIT121.

The outcomes of the assignment work are due by **Saturday 25 March 2023, 11.55 pm (sharp).**

General Java coding requirements:

- You should 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.
- You should create identifiers with sensible names.
- You should make 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: CSIT121
   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 Department, Employee and Project (DEP) in Java. This system helps a company to manage employees and projects.

Implementation

Your program shall contain at least the following classes for the DEP.
Define a class `Department` in a source file `Department.java` that contains private data members:

- `dNumber`: Department number. It is an integer type.
- `dName`: Department name. It is a String type.
- `manager`: Department manager. It is an integer type.
- `budget`: Department budget. It is a double type.
- `startDate`: Manager start date. It is a String type.

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

- `Parameterized constructor` that assigns values to all data members.
- Public method `toString` that returns the string value of all private data members. See the examples of the processing for the details of the format of the `toString` method.

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

- `eNumber`: Employee number. It is an integer type.
- `eName`: Employee name. It is a String type.
- dob: Date of birth. It is a String type.
- address: Employee address. It is a String type.
- gender: Employee gender. It is a String type.
- `salary`: Employee salary. It is a double type.
- supervisor: Supervisor number. It is an integer type.
- `dNumber`: Department number. It is an integer type.

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

- `Parameterized constructor` that assigns values to all data members.
- Public method `getNumber` that returns the employee number.
- Public method `toString` that returns the string value of all private data members. See the examples of the processing for the details of the format of the `toString` method.

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

- `pNumber`: Project number. It is a long integer type.

- title: Project title. It is a String type.
- sponsor: Project sponsor. It is a String type.
- dNumber: Department number. It is an integer type.
- budget: Project budget. It is a double type.

Implement Java methods in the file Project.java that include:

- Parameterized constructor that assigns values to all data members.
- Public method getNumber that returns the project number.
- Public method toString that returns the string value of all private data members. See the examples of the processing for the details of the format of the toString method.

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

- proj: Project object. It is a Project type.
- emp: Employee object. It is an Employee type.
- hours: Total working hours. It is an integer type.

Implement Java methods in the file WorksOn.java that include:

- Parameterized constructor that assigns values to all data members.
- Public method toString that returns the string value of all private data members. See the examples of the processing for the details of the format of the toString method.

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

- depts: A container that can store departments. It is an ArrayList of Department type.
- emps: A container that can store employees. It is an ArrayList of Employee type.
- projs: A container that can store projects. It is an ArrayList of Employee type.
- works: A container that can store objects of works on. It is an ArrayList of WorksOn type.

Implement Java methods in the file DEP.java that include:

- Default constructor that initialise values to all data members.
- Use the departments' information provided below to add the objects of departments to the correct container.
- Use the employees' information provided below to add the objects of employees to the correct container.
- Use the projects' information provided below to add the objects of projects to the correct container.
- Display a menu that includes items below.

1. Display all departments.

2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.

The Java program gets an input number of the menu and processes the required method for that item until 0 is input.

- When item 1 is selected, the method displays all the department information.
- When item 2 is selected, the method displays all the employee information.
- When item 3 is selected, the method displays all project information.
- When item 4 is selected, the method displays all information for employees who works on projects.
- When item 5 is selected, the method gets inputs of employee number, project number and total hours that the employee works on the project.
  The method verifies input information:
    - If the employee exists, then save it to an employee object variable.
    - If the project exists, then save it to a project object variable.
    - If the employee has not been allocated to the project, input the total working hours. Save the works on information to the correct container.
    - Otherwise, the method returns to the caller.

<u>Testing data</u>

The data of departments, employees and projects are given below.

Department:

```
1, "SALES", 110, 1234.00, "02/01/2012"
2, "ACCOUNTING", 120, 5566789.50, "30/10/2010"
3, "GAMES", 150, 100000.00, "01/03/2008"
4, "HUMAN RESOURCES", 200, 500000.0, "02/01/2013"
5, "SPORTS", 250, 8500000.00, "10/05/2010"
6, "RESEARCH", 300, 45500.00, "10/06/2020"
7, "EDUCATION", 350, 100000.00, "10/07/2019"
8, "FINANCE", 500, 8400000.00, "10/08/2022"
9, "COMPUTING", 600, 90000.00, "10/09/2018"
```

Employee:

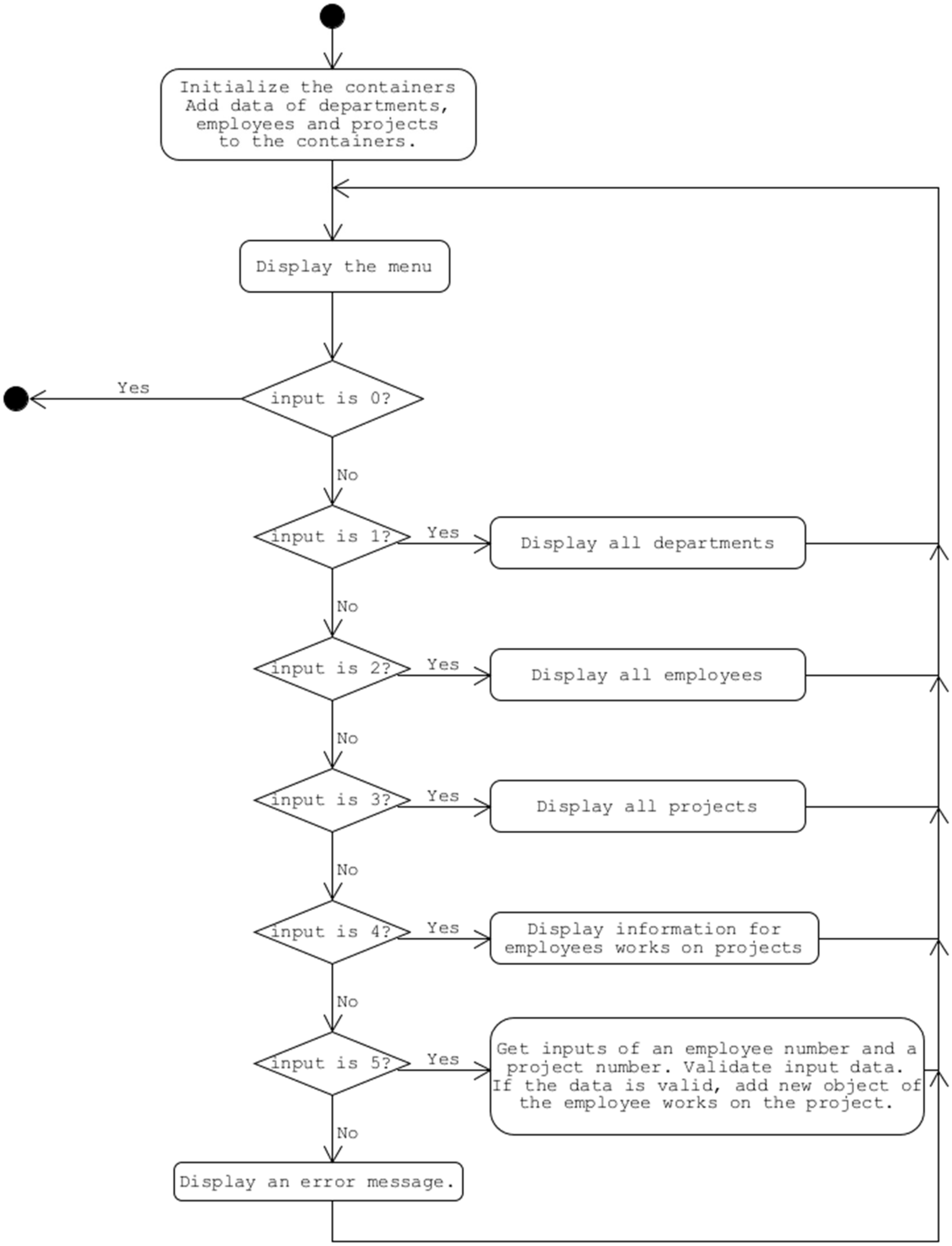
```
600, "Willy", "01/01/1988", "41 Station Street, Wollongong,
NSW 2500", "M", 250.5, 0, 9
700, "Zhi", "12/09/1999", "112 Smith Street, Windang, NSW
2525", "M", 80.2, 600, 9
800, "Mary", "03/10/2000", "26 Gibsons Road, Figtree, NSW
2525", "F", 50.0, 700, 9
```

```
500, "Angelina", "20/11/1990", "25 Bong Bong Road, Horsley,
NSW 2530", "F", 250.0, 0, 8
510, "Anna", "20/11/1990", "200 Cemetary Road, Hawea, NSW
2800", "F", 100.0, 500, 8
520, "Madelaine", "20/11/1990", "23 Lake View Street,
Figtree, NSW 2525", "F", 50.0, 510, 8
530, "Robert", "20/11/1990", "80 Penny Road, Windang, NSW
2520", "M", 50.0, 510, 8
540, "Claudio", "20/11/1990", "23 Horsley Street,
Unanderra, NSW 2528", "M", 50.0, 510, 8
350, "Brian", "13/05/1965", "23 Station Street, Wollongong,
NSW 2500", "M", 200.4, 0, 7
110, "Alvin", "13/10/1977", "56 Marlo Road, Wollongong, NSW
2500", "M", 156.4, 100, 1
```

Project:

```
1001, "Computation", "Microsoft", 8, 25000
1002, "Study methods", "Education committee", 3, 15000
1003, "Racing car", "Cloud Pty Ltd", 3, 225000
1004, "Football", "Football Club", 5, 35000
1005, "Swimming", "Education Committee", 5, 125000
1006, "Database", "Database Committee", 5, 125000
```

UML activity diagram



Compilation and testing

Compile your program by using the `javac` command.

```
      javac DEP.java
```

Process your program by using the `java` command.

```
      java DEP
```

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

Processing example

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

```
1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 1
Department number: 1, Department name: SALES, Manager number:
   110, Budget: 1234.0, Manager start date: 02/01/2012
Department number: 2, Department name: ACCOUNTING, Manager
   number: 120, Budget: 5566789.5, Manager start date:
   30/10/2010
Department number: 3, Department name: GAMES, Manager number:
   150, Budget: 100000.0, Manager start date: 01/03/2008
Department number: 4, Department name: HUMAN RESOURCES,
   Manager number: 200, Budget: 500000.0, Manager start date:
   02/01/2013
Department number: 5, Department name: SPORTS, Manager
   number: 250, Budget: 8500000.0, Manager start date:
   10/05/2010
Department number: 6, Department name: RESEARCH, Manager
   number: 300, Budget: 45500.0, Manager start date:
   10/06/2020
Department number: 7, Department name: EDUCATION, Manager
   number: 350, Budget: 100000.0, Manager start date:
   10/07/2019
Department number: 8, Department name: FINANCE, Manager
   number: 500, Budget: 8400000.0, Manager start date:
   10/08/2022
Department number: 9, Department name: COMPUTING, Manager
   number: 600, Budget: 90000.0, Manager start date:
   10/09/2018
```

```
1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 2
Employee number: 600, Employee name: Willy, Date of birth:
   01/01/1988, Address: 41 Station Street, Wollongong, NSW
   2500, Gender: M, Salary: 250.5, Supervisor number: 0,
   Department number: 9
Employee number: 700, Employee name: Zhi, Date of birth:
   12/09/1999, Address: 112 Smith Street, Windang, NSW 2525,
   Gender: M, Salary: 80.2, Supervisor number: 600,
   Department number: 9
Employee number: 800, Employee name: Mary, Date of birth:
   03/10/2000, Address: 26 Gibsons Road, Figtree, NSW 2525,
   Gender: F, Salary: 50.0, Supervisor number: 700,
   Department number: 9
Employee number: 500, Employee name: Angelina, Date of birth:
   20/11/1990, Address: 25 Bong Bong Road, Horsley, NSW 2530,
   Gender: F, Salary: 250.0, Supervisor number: 0, Department
   number: 8
Employee number: 510, Employee name: Anna, Date of birth:
   20/11/1990, Address: 200 Cemetary Road, Hawea, NSW 2800,
   Gender: F, Salary: 100.0, Supervisor number: 500,
   Department number: 8
Employee number: 520, Employee name: Madelaine, Date of
   birth: 20/11/1990, Address: 23 Lake View Street, Figtree,
   NSW 2525, Gender: F, Salary: 50.0, Supervisor number: 510,
   Department number: 8
Employee number: 530, Employee name: Robert, Date of birth:
   20/11/1990, Address: 80 Penny Road, Windang, NSW 2520,
   Gender: M, Salary: 50.0, Supervisor number: 510,
   Department number: 8
Employee number: 540, Employee name: Claudio, Date of birth:
   20/11/1990, Address: 23 Horsley Street, Unanderra, NSW
   2528, Gender: M, Salary: 50.0, Supervisor number: 510,
   Department number: 8
Employee number: 350, Employee name: Brian, Date of birth:
   13/05/1965, Address: 23 Station Street, Wollongong, NSW
   2500, Gender: M, Salary: 200.4, Supervisor number: 0,
   Department number: 7
Employee number: 110, Employee name: Alvin, Date of birth:
   13/10/1977, Address: 56 Marlo Road, Wollongong, NSW 2500,
   Gender: M, Salary: 156.4, Supervisor number: 100,
   Department number: 1

1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 3
Project number: 1001, Project title: Computation, Sponsor:
   Microsoft, Department number: 8, Project budget: 25000.0
Project number: 1001, Project title: Computation, Sponsor:
   Microsoft, Department number: 8, Project budget: 25000.0
Project number: 1002, Project title: Study methods, Sponsor:
   Education committee, Department number: 3, Project budget:
   15000.0
Project number: 1003, Project title: Racing car, Sponsor:
   Cloud Pty Ltd, Department number: 3, Project budget:
   225000.0
Project number: 1004, Project title: Football, Sponsor:
   Football Club, Department number: 5, Project budget:
   35000.0
Project number: 1005, Project title: Swimming, Sponsor:
   Education Committee, Department number: 5, Project budget:
   125000.0
Project number: 1006, Project title: Database, Sponsor:
   Database Committee, Department number: 5, Project budget:
   125000.0

1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 4

1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 5
Input an employee number: 100
The employee 100 does not exist.
```

If the employee exists

```
1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 5
Input an employee number: 500
Input a project number: 1000
The project 1000 does not exist.
```

If the project exists

```
1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 5
Input an employee number: 500
Input a project number: 1004
Input total hours: 10

1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 5
Input an employee number: 500
Input a project number: 1004
The employee 500 has already been allocated to the project
   1004
```

If the employee has the project

```
1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 5
Input an employee number: 500
Input a project number: 1005
Input total hours: 20

1. Display all departments.
```

```
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 4
Project number: 1004, Project title: Football, Sponsor:
   Football Club, Department number: 5, Project budget:
   35000.0
Employee number: 500, Employee name: Angelina, Date of birth:
   20/11/1990, Address: 25 Bong Bong Road, Horsley, NSW 2530,
   Gender: F, Salary: 250.0, Supervisor number: 0, Department
   number: 8
Hours: 10

Project number: 1005, Project title: Swimming, Sponsor:
   Education Committee, Department number: 5, Project budget:
   125000.0
Employee number: 500, Employee name: Angelina, Date of birth:
   20/11/1990, Address: 25 Bong Bong Road, Horsley, NSW 2530,
   Gender: F, Salary: 250.0, Supervisor number: 0, Department
   number: 8
Hours: 20

1. Display all departments.
2. Display all employees.
3. Display all projects.
4. Display information for employees who works on projects.
5. Add a record for an employee who works on a project.
0. Exit.
Input a choice (0-5): 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 `CSIT121` 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 `DEP.docx`.

(2) Implementation (6 marks): Implement the DEP system according to the UML class diagrams and implementation descriptions 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 items. 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 `DEP.docx`. When ready convert the Word file `DEP.docx` into a pdf file `DEP.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 **`DEP.java`**, **`Department.java, Employee.java, Project.java, WorksOn.java and DEP.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 **`CSIT121 (S123) Object Oriented Design and 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) Move a file **`DEP.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 **`Department.java, Employee.java, Project.java, WorksOn.java and DEP.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