48024 Applications Programming Assignment 1

Topics:

OO Design, Standard Patterns, Lists

Learning Outcomes:

This assessment task addresses the following subject learning objectives (SLOs): 1, 2 and 3

Due date:

11:59PM 08 October

Individual Work

All work is individual. You may discuss ideas, approaches and problems, but you should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code. You **MUST NOT** let another student see your solution code, and you MUST NOT look at another student's solution code. More information about Academic Misconduct can be found at:

http://www.gsu.uts.edu.au/rules/student/section-16.html

Specification

The STP Payroll Management System (or STP) will consist of two main components, an administrative component, and an STP component.

The administrative section will allow, through text-based menus, the login and logout of employers, the addition and removal of employees in the system, the display of employees records, the display of a certain employee with name, and the update of employees' accounts.

The STP component will store a list of all the payroll information of the employees and a list of the summary of the payrolls. The STP component will also allow the creation and display of the payroll information report based on date. Finally, the STP component will also handle the display of a certain employee payroll information.

Each employee record will include the employee's name, email, phone, address, TFN, type, paid hours and hourly rate.

Each payroll record will include the employee's name, its wages, its tax, its net and its super.

An aside

While reading the first part of the specification, you will notice there is a lot going on.

- How many functions did you identify?
- How many classes did you identify?
- What are the fields in each class?
- How many goals did you identify?
- How many patterns did you think of that might be applicable?

This assignment will be challenging, and you will probably want to manage your time well.

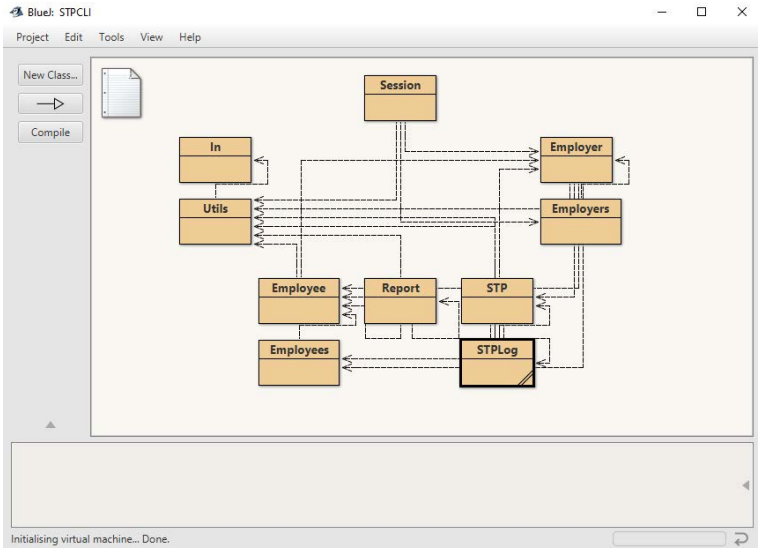
- How long do you think it will take you to code the functions?
- How long do you think it will take you to code each goal?
- A good rule of thumb is to think of an estimate, and then multiply that number by 3 or 4!
- To manage your time well, you may need to figure out which parts of the assignment you can start early.
- Which parts can you start now?
- Which parts can you start in week 6?

If you complete parts in the same week that you learn the topics (while they are fresh in your mind), they will take less time to complete.

Requirements

- Your design will consist of exactly the following classes with the listed fields, declared as indicated. You may not add or remove classes or fields; however, you may add constructors, functions and procedures to complete your design (in fact, you will have to!). You should pay careful attention to the tests on ED, as these will help guide you with some (but not all) of these methods.

- To help visualize the design, a partial class diagram has been provided



- Classes – your design will consist of these 11 classes:

  1. Session
  2. Employer
  3. Employers
  4. Employee
  5. Employees
  6. Report
  7. STP
  8. STPLog
  9. Utils (this is the class to facilitate format, do not change)
  10. In (this is just the class you've been using throughout the labs to facilitate simpler I/O)

- Fields – All the fields have been clarified in each class and they should not be modified. The fields also have some additional requirements and structures:
  Lists all have the abstract type of List<>, but must be instantiated with a concrete type that implements the List<> behavior (you can choose either – you may also want to think about why you might do things this way).

- Constructors – the constructors of the class have the following requirements:

  1. All constructors initialize the fields of their class.

2. The Session and STPLog constructors take no parameters.
3. The Employer constructor takes four parameters, the name, email, password, and employees, corresponding to the four fields identically named.
4. The Report and Reports constructors take employee as the parameter to initialize the fields identically named.
5. The STP constructor takes employer and employee as the parameters.
6. The Employee constructor requires the following formula to set the value of the fields:

   Income: hours * payPerHour*52

   Net: income – (income*rate)

   Deduction: income - net

   superRate: 0.09

   superannuation: income*superRate

   rate: 180000<=Income rate=45%

   120000<=Income <180000 rate=37%

   45000<=Income < 120000 rate=32%

   15000<=Income < 45000 rate=19%

7. Employers has the following data for Login:

John Smith	john.smith@uts.com	super123
Jane Doe	jave.doe@uts.com	user222

8. Employees has the following data.

name	email	phone	address	TFN	type	hours	payPer Hour
Thomas Muller	thomas.muller@uts.com	99991111	3 Byern St. Sydney 2001	888-888	Full-time	40	35
Alice Stefan	alice.stefan@uts.com	88881111	24 Pitt St. Sydney 2001	777-123	Part-time	20	29
Lucy Lu	lucy.lu@uts.com	98981100	11 Hunter St. Sydney 2100	111-154	Casual	20	45
Andreas Brehme	andreas.b@uts.com	90001222	91 Sussex St. Sydney 2100	172-288	Full-time	40	33
Ruddy Voller	ruddy.v@uts.com	98980000	15 Stan St. Sydney	155-154	Full-time	40	80

			2100				
Monica Shwarz	monica.s@uts.com	92241188	155 Jones St. Sydney 2001	110-994	Casual	8	20

- Utils Class – the Class defines the following formatting:

1. `System.out.format(Utils.employeeFormat, name,email,phone,type)` will produce a string of the form:

   `name,email,phone,type`

   e.g.

   `| Alice Stefan | alice.stefan@uts.com | 88881111 | Part-Time |`

   Note: "| %-20s | %-25s | %-10s | %-10s |%n" defines the space for the string.

2. `System.out.format(Utils.stpFormat,name,wages,tax,net,superannuation)` will produce a string of the form:

   `name,wages,tax,net,superannuation`

   e.g.

   `| Lucy Lu | 46800 | 14976.00 | 31824.00 | 4212.00`

   Note: "| %-15s | %-11.2f | %-11.2f | %-11.2f | %-11.2f |%n " defines the space for the string.

3. `Utils.formattedDate(this.date)` will produce a string of the form:

   `dd/MM/yyyy`

   e.g.

   `11/09/2021`

- The main method of the program will be in the Session class.

Advanced Requirements

To achieve a mark of >84, you must implement the report archive function and retrieve function (or set of functions). This corresponds to an item in the STP menu. Choosing archive option should prompt the user for a valid creation date, and then create an stpLog. Choosing retrieve option should display the stpLog on the creation date. . If the system does not archive the report on the date, system will show "No STP is recorded on:`<date>`".

Online Support

The Assignment 1 channel has been set up on ED discussion board so that students can ask questions, and other students can reply. The teaching staff will only post a reply only if the student response was wrong, or

in the case of correcting a mistake in the assignment specification.

You must not post or share Java code to the discussion board. The board is there to help you, not to provide the solution. Posting your code is academic misconduct and will reported. Each time this rule is violated, the code will be removed and replaced with a comment of the form: “Strike 1: Posting code”. After 3 strikes, you will be muted from the discussion board for the whole semester.

A dynamic FAQs (Frequently Asked Questions) has been pinned as the megathread in the channel and their answers will be posted on ED alongside the question. If you have a question, check the FAQ (megathread) first, it may already be answered there. You should read the FAQ (megathread) at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ (megathread) is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) two days before the due date; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, check the FAQ (megathread) first, then contact the subject coordinator who will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email huan.huo@uts.edu.au to ask for any clarifications or corrections to the assignment.

ED Marking

Your solution is marked for correctness by comparing the output of your system to the output of the benchmark system in ED. You can submit a solution to ED many times by press “MARK”; I urge you to do this, so you receive credit for your work. Any code hasn’t been “MARK” by ED won’t be credited.

ED will test the features of your program in a certain order, but it cannot test the more advanced goals until the basic goals are working. To receive marks, you must pass ED’s test cases in the order in which ED tests them.

Your code is marked by software, so you can get a good mark by fooling or spoofing the software. If you spoof a task worth N marks, you receive a penalty of 2*N marks.

Submission and Return

Your provisional mark and feedback is generated immediately each time you submit to ED. However, it takes time for the analysis of spoofing, plagiarism, collusion and general cheating, which will start two weeks following the due date. If you are suspected of Academic Misconduct, I will forward your case to the Misconduct Committee and will notify you by manual feedback in ED.

There is no scheduled late submission period. An extension of up to 3 days may be given by the subject coordinator before the due date; you have to supply documentary evidence of your claim. An extension CANNOT be given after the due date or exceed 3 days. You may also apply for special consideration for reasons including unexpected health, family or work problems. More information about how to apply for special consideration can be found at: http://www.sau.uts.edu.au/assessment/consideration.html.

Marking Scheme

The marks for the assignment are divided into the following functionality components (note that individual tests may test several functionality components, and a functionality component may be tested by several tests):

Functionality Component	Mark Allocation
Main Menu	4
Employer Menu	5
STP Menu	5
View Employee	10
Add Employee	10
Retrieve Employee	10
Update Employee	10
Delete Employee	10
PAYG Report	10
STP Report	10
Archive Report	6
Retrieve Report	10

This adds to a mark out of 100, at makes up 35% of your final assessment mark.