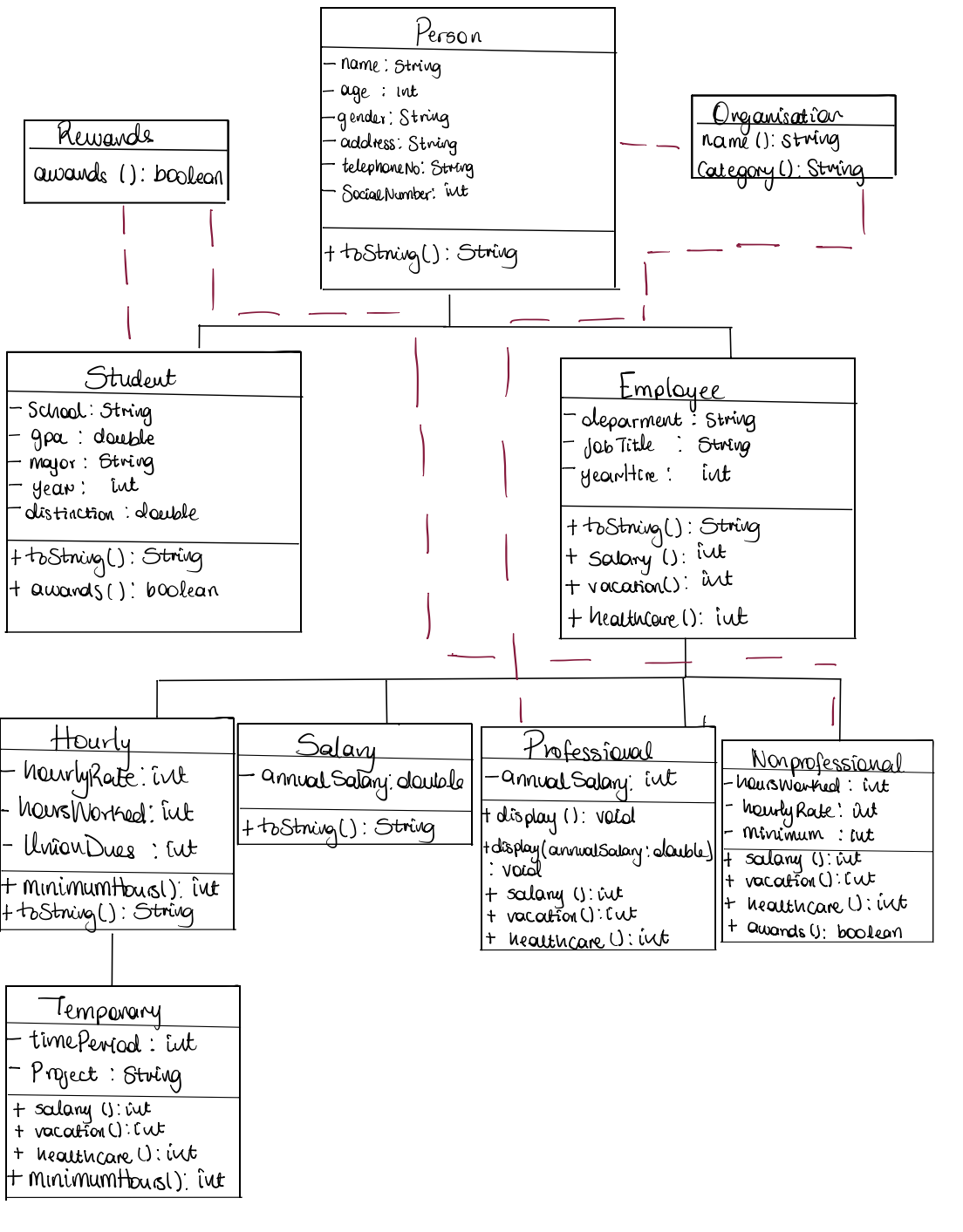
Maintaining code and software engineering guidelines

Obuwoma oghenekeno

Department, University of Ottawa

ITI 1121 Report

13th June,2024



b. Section where you discuss how you improved on the design and organization from assignment 1.

From assignment 1, I have improved in the organization of my code in numerous ways such as putting all of my classes in separate files which has made my code easier to understand. In the previous design, my system did not have any interfaces so all the relationships were created from inheritance but in this new implementation I have added two new interfaces which have increased the flexibility and reusability of my system. Moreover I ensured my interfaces were relevant to all the other classes so it could be used by them.

c. Section where you discuss the different polymorphism examples in your implementation.

Which include:

1. Class student and NonProfessional implementing the awards method in the reward interface: Both classes implement the interface rewards so therefore they must implement all the methods contained in the reward interface, in this cases the reward interface only contain one method called awards which of boolean type and it is meant to return true if the individual meets the requirements to get an award. The reward interface is reusable because most organizations give awards to their employees or students.
2. Class Temporary overriding minimum hours in class Hourly: The class Temporary is a child class of the class Hourly, the class Hourly is an abstract class which declares an abstract method minimumHours and since Temporary is inheriting from Hourly, it has to provide an implementation for minimumHours for it to run. The method minimumHours is intended to calculate the minimumHours a temporary(seasonal) employee is expected to work in a week depending on the duration time period their contract is valid for.
3. Class professional has an overloaded method display: The display method has two different signatures which are determined during run time. The first form is empty and returns only the information that applied to Person from the class Professional while the second form contains a parameter called annual Salary which overrides the previous annual salary and displays the information which applies to Professional.
4. Class Person and Professional implement Interface Organization: Both classes implement the interface Organization so therefore the classes both implement the methods in Organisation, in this case the organization interface contains two methods name and category which allows you to group individuals based on what category you want. For example in the Professional class it returns the department of the employee and their name while in the class Person it returns the name and gender of the person.

d. Section where you discuss the lessons you learned from assignments 1 and 2.

From assignment 1 and 2, I have been able to grasp concepts that where challenging to understand like:

1. Understanding polymorphism : I have been able to comprehend the need for polymorphism and how it makes the system we design more maintainable and reusable. I learnt that the method toString was a method in the object class which has to be overridden to display the proper string representation and that when a class implements an interface it implements its method which is considered as polymorphism and also how to overload a method.
2. Understanding inheritance: In the beginning I thought that once a class inherits from another class that the subclass could not be a parent class for another subclass but now I know that that's wrong. For instance, in my system class Hourly extends employee but Temporary extends Hourly as well so although multiple inheritance is not allowed, class hierarchy is allowed.
3. Understanding abstract classes: The concept of abstract class was not clear to me or why have abstract class when they are interface and concrete method but now i understand that abstract classes contain but concrete and abstract method and they can be used to enforce their subclasses to create implementations of abstract methods and while they cannot be instantiated, they can store instances of their subclasses if they are not abstract as well.
4. Different between instance variables and class variables: From class, I understood that Instance variables are unique to every single instance of a class while class variables are shared across all instances of a class but i didn't fully understand how they were access or called, i would constantly mix them up but now i understand that only instance method can access instance variables so you have to create instance to actually use the method but for class methods you can just access it with the class.