**Lab assignment**: Payroll System Using Inheritance and Polymorphism

In this lab you will get experience with some of the implementation issues and conceptual details of inheritance. Inheritance is a mechanism for increasing the reusability and reliability of C++ code. It is worth mentioning that inheritance is a characteristic of all object-oriented programming languages. Our goal is to give you a glimpse of the functionality of inheritance, so that you can make informed design decisions in the future. Please read through the entire lab before you begin. You can refresh your memory on inheritance by reviewing the lectures!

Lab Insight

This lab teaches some of the characteristics observed in object-oriented programming. These characteristics are useful in software design when building code bases for large projects, APIs, and production code bases. Some classes that further extend on these concepts. This lab will help you build clean code bases where the OOP (Object-Oriented Programming) characteristics help minimize having to rewrite redundant code as well as make use of virtual inheritance to treat custom sub-classes the same way as their base classes without losing the custom functionality of these subclasses.

## Steps

1. Define the following constants in a header file.

* FACULTY\_MONTHLY\_SALARY = 5000.00
* STAFF\_MONTHLY\_HOURS\_WORKED = 160

1. Implement an abstract class Employee with the following requirements:

* Attributes
  + last name (String)
  + first name (String)
  + ID number (String)
  + Sex - M or F
  + Birth date (Date)
* Default argument constructor and argument constructors.
* Public methods
  + putData that displays the following information:  
    ID Employee number :\_\_\_\_\_\_\_\_\_  
    Employee name: \_\_\_\_\_\_\_\_\_\_  
    Birth date: \_\_\_\_\_\_\_
  + get and set methods
  + pure virtual method monthlyEarning that returns the monthly earning.

1. Implement a class called Staff extending from the class Employee with the following requirements:

* Attribute
  + Hourly rate
* Default argument and argument contructors
* Public methods
  + get and set
  + The method monthlyEarning returns monthly salary (hourly rate times 160)
  + putData that displays the following information:  
    ID Employee number :\_\_\_\_\_\_\_\_\_  
    Employee name: \_\_\_\_\_\_\_\_\_\_  
    Birth date: \_\_\_\_\_\_\_  
    Full Time  
    Monthly Salary: \_\_\_\_\_\_\_\_\_

1. Implelment a class Education with the following requirements:

* Attributes
  + Degree (MS or PhD )
  + Major (Engineering, Chemistry, English, etc ... )
  + Research (number of researches)
* Default argument and argument constructors.
* Public methods
  + get and set

1. Implement a class Faculty extending from the class Employee with the following requirements:

* Attributes
  + Level  
    "AS": assistant professor  
    "AO": associate professor  
    "FU": professor
  + Education object
* Default argument and argument constructor
* Public methods
  + get and set
  + The method monthlyEarning returns monthly salary based on the faculty's level.  
    AS - faculty monthly salary  
    AO - 1.2 times faculty monthly salary  
    FU - 1.4 times faculty monthly salary
  + putData that displays the following information:  
    ID Employee number :\_\_\_\_\_\_\_\_\_  
    Employee name: \_\_\_\_\_\_\_\_\_\_  
    Birth date: \_\_\_\_\_\_\_  
    XXXXX Professor where  XXXXX can be Assistant, Associate or Full  
    Monthly Salary: \_\_\_\_\_\_\_\_\_

1. Implement a class called Partime extending from the class Staff with the following requirements:

* Attributes
  + Hours worked per week
* Default argument and argument constructors
* Public methods
  + set and get
  + The method monthlyEarning returns monthly salary which hourly rate multiplied hours worked per week multiplied four.
  + putData that displays the following information:  
    ID Employee number :\_\_\_\_\_\_\_\_\_  
    Employee name: \_\_\_\_\_\_\_\_\_\_  
    Birth date: \_\_\_\_\_\_\_  
    Hours works per month: \_\_\_\_\_\_  
    Monthly Salary: \_\_\_\_\_\_\_\_\_

1. Implement a test driver program that creates a one-dimensional array of class Employee to store the objects Staff, Faculty and Partime.  
     
   Using polymorphism, display the following outputs:  
     
   a. Employee information using the method putData.

* All employees
* Staff
* Faculty
* Part-time

b. Total monthly salary for all the part-time staff .  
 c. Total monthly salary for all employees.

Test Data

Staff

1. Last name:  Allen  
   First name: Paita  
   ID: 123  
   Sex: M  
   Birth date: 2/23/59  
   Hourly rate: $50.00
2. Last name: Zapata  
   First Name: Steven  
   ID: 456  
   Sex: F  
   Birth date: 7/12/64  
   Hourly rate: $35.00
3. Last name:Rios  
   First name:Enrique  
   ID: 789  
   Sex: M  
   Birth date: 6/2/70  
   Hourly rate: $40.00

Faculty

1. Last name: Johnson  
   First name: Anne  
   ID: 243  
   Sex: F  
   Birth date: 4/27/62  
   Level: Full  
   Degree: Ph.D  
   Major: Engineering  
   Reseach: 3
2. Last name: Bouris  
   First name: William  
   ID: 791  
   Sex: F  
   Birth date: 3/14/75  
   Level: Associate  
   Degree: Ph.D  
   Major: English  
   Reseach: 1
3. Last name: Andrade  
   First name: Christopher  
   ID: 623  
   Sex: F  
   Birth date: 5/22/80  
   Level: Assistant  
   Degree: MS  
   Major: Physical Education  
   Research: 0

Part-time

1. Last name: Guzman  
   First name: Augusto  
   ID: 455  
   Sex: F  
   Birth date: 8/10/77  
   Hourly rate: $35.00  
   Hours worked per week: 30
2. Last name: Depirro  
   First name: Martin  
   ID: 678  
   Sex: F  
   Birth date: 9/15/87  
   Hourly rate: $30.00  
   Hours worked per week:15
3. Last name: Aldaco  
   First name: Marque  
   ID: 945  
   Sex: M  
   Birth date: 11/24/88  
   Hourly rate: $20.00  
   Hours worked per week: 35

**GRADING**

* Inside the program, put your name at the beginning of your program  
  //YOUR NAME  
  // KAU-ADVP- 2022
* Return a hard copy of the lab assignment
* Demonstrate the lab assignment in the lab
* Grade based on the successful execution of the program and the coding efficiency.