INHERITANCE

The key topic for this week is introduction to key concepts of OOP, inheritance.

TASK

1. Practice defining subclasses
2. Practice polymorphism via inheritance
3. Understand the inheritance hierarchy
4. Practice defining and generating UML diagram

Part A

1. True or false:
   a. A class can directly extend more than one class.
   b. Over-riding code in a subclass must have exactly the same parameters and return type as the method in the superclass.
   c. The subclass object inherits the fields and methods of the superclass, but if you want to use them in the added code in the subclass, you must make the fields and methods in the superclass either `protected` or `public`.

2. Consider classes `Telephone`, `MobilePhone`, `SatellitePhone`, `Samsung`, `IPhone`,`IPhone6S` and `Nokia.` Draw an UML diagram to illustrate the inheritance relationships between these classes; and sketch the code for some of the instance fields, constructors, and methods for each of these classes for better code reuse.

Part B

Exercise 1:

Write a superclass `Worker` and subclasses `HourlyWorker` and `SalariedWorker`. Every worker has a name and a salary rate. Write a method `computePay (int hours)` that computes the weekly pay for every worker.

An hourly worker gets paid the hourly wage for the actual number of hours worked, if hours are at most 40. If the hourly worker worked more than 40 hours, the excess time is paid at 150%. The salaried worker gets paid the hourly wage for 40 hours, no matter what the actual number of hours is.

Exercise 2:

Implement a superclass `Person`. Make two classes `Student` and `Instructor` that inherit from `Person`. A person has a name and a date of birth. A student has a major, and an instructor has a salary. Write the class definition, the constructors, and the methods `toString` for all classes. Supply a test program that tests these classes and methods.

Exercise 3: (download BankAccount.java and SavingsAccount.java from course website)

Add a `TermDepositAccount` class to the bank account hierarchy. The term deposit account is just like a savings account, but you promise to leave the money on the account for a particular number of months, and there is a penalty for early withdrawal. Construct the account with the interest rate and the number of months to maturity. In the `addInterest` method, decrement the count of months, if the count is positive during a withdrawal, charge the withdrawal penalty. Supply a test program.