Final Assignment (Theory)
Object Oriented Programming
Mark 05
Please submit the printed copy in my office.
If any plagiarism is occurred, all will get zero.
***Deadline 30 April 2023**

A University has different types of employees such as Professors, Administrators, and Support Staff. Each employee has a unique ID, name, and salary. Professors also have a field for their subject of expertise, administrators have a field for their department, and support staff have a field for their job title.

Your task is to create a class diagram (UML) and Java program that models these employees using Object-Oriented Programming concepts. You should create a superclass called "Employee" and then create three subclasses called "Professor," "Administrator," and "SupportStaff." Each subclass should inherit from the "Employee" superclass.

The "Employee" superclass should have the following attributes and methods:

Attributes:

- employeeID (a unique identifier for each employee)
- name (the name of the employee)
- salary (the salary of the employee)

Methods:

- getEmployeeID() (returns the employeeID)
- getName() (returns the name)
- getSalary() (returns the salary)
- toString() (returns a string representation of the employee)

The "Professor" subclass should have the following additional attribute and method:

Attribute:

- subjectOfExpertise (the subject that the professor is an expert in)

Method:

- getSubjectOfExpertise() (returns the subjectOfExpertise)
- toString() (overrides the superclass method to include the subjectOfExpertise)

The "Administrator" subclass should have the following additional attribute and method:

Attribute:

- department (the department that the administrator works in)

Method:

- getDepartment() (returns the department)
- toString() (overrides the superclass method to include the department)

The "SupportStaff" subclass should have the following additional attribute and method:

Attribute:

- jobTitle (the job title of the support staff)

Method:

- getJobTitle() (returns the jobTitle)
- toString() (overrides the superclass method to include the jobTitle)

Your program should create at least one instance of each subclass, and then print out the details of each employee using the toString() method.

Additionally, your program should implement a method called "totalSalary" that takes an array of employees and returns the total salary of all employees in the array. This method should be implemented in the "Employee" superclass, and should use polymorphism to calculate the total salary based on the type of each employee.

For example, if the array contains two professors, one administrator, and one support staff, the total salary should be the sum of the salaries of the two professors, the administrator, and the support staff.