

# Programming Assignment 2A

## HR Management System

### Introduction

The HR system is a software application that aims to automate the management of employee records, benefits, and payroll. The system is designed to be modular, with several classes such as Employee, Benefits, Payroll, and Department, each representing a distinct area of functionality. The main objective of the HR system is to provide an efficient and user-friendly way for HR personnel to manage employee data, benefits, and compensation.

### Instructions:

1. Write an application to implement "**HR management system**" as described below.
2. The solution is to be presented in a **report**. This report is to contain
  - o A through description of the application,
  - o Classes Diagram,
  - o A description of underlying data structures used to solve the problem,
  - o A copy of the C++ code,
  - o A description of all the function/routines which have been used, and
  - o Test data used and sample execution screenshots of output produced.
3. A copy of the code and an executable is to be placed in a folder called **OOP2a**. A copy of the code must also be uploaded on teams. You will be required to demo your application for assessment after the due date.

## Details

### Classes

There should be the following classes:

- Employees classes:** This base class stores the basic information of an employee, including their name, ID, job title, phone, email, and a list of benefits. There are four types of employees. The Hourly/Salaried/Commission and Manager is an Employee with different attributes and a new equation for payroll function as shown in the starter diagram. The following table shows the payroll equation. The getDetail function should customize the output based on employee type.

Class	Calculate Payroll
Hourly Employee	rate*hours
Salaried Employee	salary
Manager Employee	salary+bonus
Commission Employee	rate * target

- Department:** This class stores information about a department, including the department name, ID. Each employee should belong to the department.
- Benefits:** This base class is to determine different types of benefits for employees. Mainly two type of benefits, the following table shows the calculation equation of benefits

Class	Calculate Benefits Amount
Health Benefit	20% of amount
Dental Benefit	10% of amount

- HRSystem:** This class is the main class that stores a list of employees and manages the operations of adding, removing, and searching for departments and employees. It should also have a method to display the information of a specific employee or the average salary of a department.

## Main Function Menu

The main function should have a menu that allows the user to perform the following operations:

- **Employee Information Management:** The system allows HR personnel to add, delete, and modify employee information, including personal information, job information, and benefits information.
- **Benefits Management:** The system allows HR personnel to manage benefits information, including adding, deleting, and modifying benefits packages.
- **Salary Calculation:** The system calculates salaries for employees based on their job type and any applicable benefits.
- **Reporting:** The system generates reports on employee information, benefits information, and salary information.
- **Search:** The system allows HR personnel to search for employees based on various criteria, including name, job title, and benefits package.
- **NOTE:** Add all requirement functions you may need such as get/set or other.

## Test Plan

- The test plan should test the functionality of all the methods in the classes, including adding, removing, and searching for departments and employees.
- The test plan should also check the calculation of the average salary of the department.

## Documentation Report

The documentation report should include the following sections:

- **Introduction:** Explain the purpose of the project and describe its functionality.
- **Design:** Outline the classes and their attributes, methods, and relationships.
- **Implementation:** Discuss the specific coding details, including code snippets and explanations.
- **Testing:** Detail the test plan and the results of the tests.
- **Conclusion:** Summarize the results of the project and suggest areas for improvement.

The report should be well-organized, with clear and concise writing. You can use headings, subheadings, and bullet points to make it easier to read.

## Starter Class Diagram

