UNIVERSITY OF TECHNOLOGY, JAMAICA
School of Computing and Information Technology
Object-Oriented Programming Project

C. Panther / R. Clarke/A. Bowen-Mighty/T. Edwards/G. Sawyers
Group Assignment (3-5 persons Per group)

Given Date: October 9, 2023 **Due Date: November 11, 2023**

Software Solutions Now (SSN) is a software development company specializing in the development of new programs and the customization of off the shelf applications. SSN was started as a small freelancing venture by two friends, the small venture has grown into a thriving business with over one hundred employees. Over the years, the company has maintained a simple payroll system in a spread sheet application, which was sufficient based on the number of employees. Now that the business has grown, the system is no longer suitable, and as such the manager of the Accounts department has requested that a new system be developed. The manager provided the following information to the Software Engineering Department, to aid in the development of the SSN Payroll Management System (PMS).

Each employee is assigned to a department and is paid based on the rates setup for each department; these rates are stored in a department rates file (the structure of the rates file is shown below). These rates are hourly rates. The overtime rate is applied once an employee works more than 40 hours.

Department Rates File (Sample Data)

Dept. Code	Dept. Name	Regular Rate $	Overtime Rate $
1001	Human Resource Management	1500.97	750.49

A separate employee payroll file is maintained that stores the data on each employee (the structure of the file is shown below), both files (Rates File & Payroll File) are tab delimited text files, with the respective column headings.

NB: Sample Rates and Payroll files will be provided, for development purposes.

Employee Payroll File (Sample Data)

ID. No	First Name	Last Name	Dept. Code	Position	Hours Worked
97015	Kerisha	Sinclair	5001	Manager	35.50

The processing of the payroll leads to the generation of a processed payroll file (the structure of the file is shown below).

Processed Payroll File (Sample Data)

ID. No	First Name	Last Name	Dept. Code	Position	Hours Worked	Regular Pay	Overtime Pay	Gross Pay
97015	Kerisha	Sinclair	5001	Manager	35.50	53,250.00	0.00	53.250.00

Program Requirements:

1. Perform an Object-Oriented Analysis (OOA) on the proposed SSN Payroll Management System (PMS) described above. The OOA should clearly show the steps used to identify potential classes and the selection of the actual classes. Based on the OOA, create an Object-Oriented Design (OOD) using the Unified Modelling Language (UML). The OOD should show appropriate UML diagrams for all classes and the class relationship diagram, should show all relationships existing between the classes.

2. Using C++ or Java, implement the SSN Payroll Management System (PMS) as your group project.

3. The user should maintain the Department Rates data for each department via a menu with the options:
    i. Add: which allows the user to add new department rates record to the system.
    ii. Update: which allows the user to update an existing department record.
    iii. View: which allows the user to view a single department record.
    iv. View All: which allows the user to view all department records.

4. The user should maintain the Employee data via a menu with the options:
    i. Add: which allows the user to add a new employee record to the system.
    ii. Update: which allows the user to update an existing employee record.
    iii. View: which allows the user to view a single employee record.
    iv. View All in Department: which allows the user to view all employee records for a specified department.
    v. Delete: which allows the user to delete an employee record.

5. The user shall process the Employee Payroll via a menu with the options:
    i. Process Payroll: Calculates payroll and generates Processed Payroll File.
    ii. View Payroll: Allows the user to view a single employee payroll record.
    iii. View Department Payroll: Allows the user to view all employee payroll records for a specific department.
6. When the exit option is selected the application should close.

7. All committed changes made during the execution of the program, should be stored and used to update the relevant files, when the application terminates.

8. The analysis of the current system also uncovered details related to the various records held in several files; all employees must provide their Taxpayer Registration Number, National Insurance Scheme number and date of birth. The company also records the date the employee was hired. Each processed payroll record should also show the current payroll date and a unique system generated cheque number.

Grading Scheme (100 marks): General Mark Breakdown

No.	Section/Objective	Max. Mark(s)	Act. Mark(s)
1.	**DOCUMENTATION**		
1.1	Signed Authorship forms (i.e. one per group member)	2	
1.2	Group Report (Outlining contribution of each group member)	3	
1.3	Object-Oriented Analysis and Design of system	10	
1.4	User Manual (Outline of how the program works including instructions to how to setup the program. This should be properly done to resemble a program already on the market)	5	
	NB: Missing Authorship Forms and Group Report will result in loss of 15 marks.		
	DOCUMENTATION SECTION TOTAL	**20**	
2.	**SOURCE CODE**		
2.1	Comments		
2.1.1	Each File(class) should have details for the students who wrote the file.	1	
2.1.2	Practice use of self-commenting files (i.e. proper variable and method naming convention taught in the module)	1	
2.1.3	Proper use of inline and method comments where necessary	1	
2.2	Naming Convention		
2.2.1	Pascal Case should be used for naming classes	2	
2.2.2	Camel Case should be used for variable and method naming	4	
2.2.3	Ensure class Files are named appropriately as per instructions	1	
2.3	Object-Oriented Programming Techniques		
2.3.1	Implementation of Inheritance in the program	3	
2.3.2	Implementation of Composition in the program	3	
2.3.3	Implementation of Method Overriding and Overloading in the program	6	
2.3.4	Implementation of Polymorphism in the program	3	
2.4.	Use of Files		
2.4.1	Proper Implementation of appropriate file management in the program	5	
	SOURCE CODE SECTION TOTAL	**30**	
3.	**FUNCTIONALITY**		
3.1	Robustness of Program		
3.1.1	User Input validation checks should be seen where required	2	
3.1.2	Implementation of Error Checks/ Exception Handling in the program	3	
3.1.3	Clearly show how the Program Navigates (using consistent menus throughout)	2	
3.2	Graphical User Interface		
3.2.1	Ease of User Interaction	6	
3.2.2	Appropriate Notifications (i.e. error and information messages)	4	
3.3	System Functionality Implementation		
3.3.1	Maintain Department Rates data	10	
3.3.2	Maintain Employee data	15	
3.3.3	Process Employee Payroll data	8	
	FUNCTIONALITY SECTION TOTAL	**50**	
	FINAL PROJECT MARK	**100**	

Extra Marks (10 marks):
A project that satisfies the program's functional requirements can gain additional marks up to a maximum of 10 marks by implementing the additional requirements as follows:

- + 5 marks – Awarded for use of extended ASCII codes and function keys (e.g., F1 – F12, Arrow Keys).
- + 5 marks – Awarded for use of colour and graphics to enhance the look and feel of the program.

Final Submission:

- Signed authorship forms, Group Report, Object-Oriented Analysis and Design (OOA&D),
- Source Files and Executable to your Lab Tutor

.

Late Submission:

- Any project submitted after the due date will be late and 10 % will be deducted for each day late.
- Late projects will not be considered for extra marks, if extra marks are available.