CSADPRG MCO Specifications

Specifications for Comparative Analysis on Programming Languages

1. Activity Description

The activity serves to support the theoretical foundations of programming languages discussed in class, by surveying, comparing, and evaluating the features and constructs of different programming languages. For this iteration of the MCO, all groups are tasked to study Go, Kotlin, R, and Ruby. The project has three components, namely a language evaluation paper, a running application software, and a class presentation. By the end of the project, students should have acquired an understanding of the different paradigms supported by various languages and appreciate the similarities and differences in their language features.

2. Software Application –Weekly Payroll System

Manual payroll computation is prone to human errors. Create an automated weekly payroll system that will generate the **pay of an employee for the week.**

A normal work week of an employee is 5 days; and a normal workday of an employee is 8 hours. However, there are times when an employee works overtime or on holidays. This makes the computation of daily salary quite complex.

Rules for salary computation.

- If the regular work hours for an employee is 8 hours per day, then a regular work schedule that starts at 0900 will end in 1800. This includes an unpaid one continuous hour of break time.
- Work from 2200 to 0600 is considered night shift. An additional 10% of the hourly rate is given for every hour of work during the night shift. (Night shift differential).
- Hourly rate is computed as daily salary divided by maximum regular work hours per day.
- If the number of days of work of an employee is 5 days, the remaining 2 days are his rest days.
- If an employee works on holidays, or on his rest days (on the 6th or 7th day of the week), he will receive additional pay for those days. Here are the rates, assuming his work hours did not exceed 8 hours.

Day	**Rate**
Rest Day	130%
Special Non-Working Day	130%
Special Non-Working Day and Rest Day	150%

Regular Holiday	200%
Regular Holiday and Rest Day	260%

- If an employee's work hours exceed 8 hours, he is paid overtime for every hour after that.

Day	Non-Night Shift	Night Shift
Normal Day	125%	137.5%
Rest Day	169%	185.9%
Special Non-Working Day	169%	185.9%
Special Non-Working Day and Rest Day	195%	214.5%
Regular Holiday	260%	286.0%
Regular Holiday and Rest Day	338%	371.8%

Configuration

The system has the following default settings:

- Daily salary is set to **500.**
- Maximum regular hours of work per day is 8 hours.
- There are 5 workdays in a week; i.e. the first 5 days are work days and the last 2 days are rest days. Rest days are continuous days at the end of the week.
- All 7 days of the week are initially normal days.
- IN time of each of the 7 days is 0900.
- OUT time of each of the 7 days is 0900 (same as the start time). These configuration settings may be changed by the user.

Assumptions

For this system, the following assumptions are true.

- Employees are not late.
- Employees work for at least 8 hours.
- Employees may be absent.

Weekly salary computation

To generate the payroll of an employee, the user will update the OUT time of each employee for each day of the week. The output time is provided in military time format.

For rest days or days when the employee is absent, OUT time entered will be 0900 (start time of a regular workday) to mean that the employee did not come in for work on this day.

For rest days, employees get regular salary, unless they worked on that day. When an employee is absent on a normal day, he will not receive any pay for that day.

The salary of the employee for the week is the sum of his computed salary for each day of the week.

System Features

- Your payroll system must be menu-driven. Users will choose from a list of options on what tasks he wants to execute. This includes exiting from the system.
- The user should be allowed to modify default configuration.
- When generating the payroll, salary for each day, and the total salary for the week must be shown.

Sample Computations:

Sample 1	
Daily Rate	500.00
IN Time	0900
OUT Time	1800
Day Type	Normal Day
Hours Overtime (Night Shift Overtime)	0 (0)
Salary for the day	**500.00**

Sample 2		
Daily Rate		500.00
IN Time		0900
OUT Time		2100
Day Type		Normal Day
Hours Overtime (Night Shift Overtime)		3 (0)
Salary for the day		**734.38**
Computation:		
Daily Rate	0900 – 1800	500.00
Hours OT × OT Hourly Rate = 3 × 500 ÷ 8 × 1.25	1800 – 2100	234.38

Sample 3		
Daily Rate		500.00
IN Time		0900
OUT Time		0000
Day Type		Normal Day
Hours Overtime (Night Shift Overtime)		4 (2)
Salary for the day		**984.38**
Computation:		
Daily Rate	0900 – 1800	500.00
Hours OT × OT Hourly Rate = 4 × 500 ÷ 8 × 1.25	1800 – 2200	312.50
Hours OT × NS-OT Hourly Rate = 2 × 500 ÷ 8 × 1.375	2200 – 0000	171.88

Sample 4		
Daily Rate		500.00
IN Time		1800
OUT Time		0300
Day Type		Normal Day
Hours on Night Shift		5
Salary for the day		**843.75**
Computation:		
Daily Rate	1800 – 0300	500.00
Hours on NS × Hourly Rate × NSD = 5 × 500 ÷ 8 × 1.10	2200 – 0300	343.75

Sample 5		
Daily Rate		500.00
IN Time		0900
OUT Time		1800
Day Type		Rest Day
Hours Overtime (Night Shift Overtime)		0 (0)
Salary for the day		**650.00**
Computation:		
Daily Rate × Rest Day = 500 × 1.30	0900 – 1800	650.00

Sample 6		
Daily Rate		500.00
IN Time		0900
OUT Time		2300
Day Type		SNWH, Rest Day
Hours Overtime (Night Shift Overtime)		4 (1)
Salary for the day		**1,371.56**
Computation:		
Daily Rate × SNWH-Rest Day = 500 × 1.50	0900 – 1800	750.00
Hours OT × OT Hourly Rate = 4 × 500 ÷ 8 × 1.95	1800 – 2200	487.50
Hours OT × NS-OT Hourly Rate = 1 × 500 ÷ 8 × 2.145	2200 - 2300	134.06

3. Evaluation Paper

The language evaluation paper contains a detailed discussion of the features and constructs of the Ruby, R, Kotlin and Go. Using code snippets, the research paper also includes an evaluation of the language with respect to the theories and concepts discussed in class. One section should discuss how specific features of a programming language aided in the implementation of the Weekly Payroll System. The required contents for the evaluation paper vary among languages, depending on the programming paradigm. The outline of the paper is given in Section 5.

Format

Margin:	1 inch margin (all sides)
Font:	12pt Times New Roman for discussion
	11pt Bold Courier New for code segments
Paper:	Short (8”x11”)
Spacing:	Single (except between paragraphs)

IMPORTANT!
Groups that simply “copy and paste” materials from different sources will automatically be given a grade of 0 for project. The contents of the paper are dependent on the source code of your specific application. Concepts and theories may be paraphrased from reference materials, but must be properly cited. Documents with no citations will receive deduction points.

Citation
We will follow the American Psychological Association (APA) format for citations. In the APA format, the author-date method of citation is being followed. This means that the author's last name and the year of publication for the source should appear in the written report, and a complete reference should appear in the reference list.

Examples:
Smith (1970) compared reaction times . . .
In a recent study of reaction times (Smith, 1970), …
In 1970, Smith compared reaction times . . .

4. Grading Scheme

Software Application	35 pts
Language Evaluation Paper	45 pts

Class Presentation	20 pts.

Language Paper Evaluation (35 pts)

1. Completeness (Did you cover everything?) 25pts
2. Relevance (Did you cover the right thing?) 10pts
3. Clarity and Organization 10pts

Class Presentation (20 pts)

1. Significance of the Discussion 10pts
2. Clarity and Organization of the Presentation 10pts

Each group is alloted 20 minutes to present its software application and programming language in class. The goal of this exercise is to provide a venue for sharing new knowledge regarding the various languages and paradigms.

5. Project Schedule and Deliverables

Language Evaluation Paper **April 6, 2023 11:59 PM**

The evaluation paper must be submitted on Canvas by 1159. Late submissions will not be accepted, and the group will automatically be given a grade of 0 for the paper. The evaluation paper should have the following contents:

· Title Page

An Evaluation Paper on
<Programming Languages>
:
submitted
in partial fulfilment of the requirements
for the course CSADPRG
:
<name of member – 1 per line, arranged in alphabetical order based on surname>
:
<CSADPRG teacher>
<date>

1. Introduction

In this section, introduce the programming paradigms of the different languages. Present an overview of the languages – history, revisions, current state, future directions. Relate them with each other. Discuss the application domain where your language is most suited for.

2. Language Comparison

   In this section, discuss the the major features of the different languages. Explain as well how the features (capabilities and constraints) of the languages can make the software development process easier and/or more difficult. Provide code snippets to clarify the discussion.

3. Developing the Weekly Payroll System using <Programming Language>

   In this section, introduce your software application by highlighting its major functions. Discuss how the chosen language of implementation made it easier and/or hindered the development.

6. Conclusion

   In this section, summarize what you have learned. Say something about the languages, share your learning experiences, explain if your application is appropriate for the languages, and other recommendations you may have.

References

The following are examples of correct listing of references.

Online Material
Author, A. A. (Date of Publication or Revision). Title of full work [online]. Available: full web address. (Date of access).

* Date of access" should indicate the date you visited the website. This is important because online information is frequently altered.

Sample 1:
Daly, B. (1997). Writing argumentative essays. [online]. Available: http://www.eslplanet.com/teachertools/argueweb/frntpage.htm. (May 12, 1998)

Sample 2
The Evolution of Truth: Phi & Fibonacci Anomalies [online]. Available: http://evolutionoftruth.com/fiv/fibomyst.htm. (September 7, 2004).

Book
Calfee, R. C., & Valencia, R. R. (1991). APA guide to preparing manuscripts for journal publication. Washington, DC: American Psychological Association.

An article or chapter of a book
O'Neil, J. M., & Egan, J. (1992). Men's and women's gender role journeys: Metaphor for healing, transition, and transformation. In B. R. Wainrib (Ed.), Gender issues across the life cycle (pp. 107-123). New York: Springer.

Class Presentation **April 6 & 13, 2024**

Submit a copy of your slides by **April 5, 2024**. **11:59 PM** The class presentations will be on April 6 & 13, 2024.

The presentation must focus on the following items:

April 6 (More details to follow)

- Overview of the programming paradigm
- Brief history of the programming language
- Source code discussion, to showcase the important features and capabilities of the programming languages (Note that only important features and capabilities that implement your data structures and algorithms must be presented in class!)

April 13

- Software demo