**Assignment-3**

**Program-1**

This lab is based on the material studied so far, including the exercises done in class.

This assignment is to be done individually; Copying or cheating Material will make you held responsible for breaking Academic dishonesty Policy.

A program is required to calculate the amount of money to pay an employee at Sheridan. There are Four types of employees:

|  |  |
| --- | --- |
| **Employee Type** | **Method of Payment** |
| Full-Time | annual salary / 44 |
| Part-Time | hours worked \* rate of pay, where the rate of pay is $30.00/hr for employees |
| Sessional | hours worked \* rate of pay, where the rate of pay is $32.00/hr for employees |
| Partial | hours worked \* rate of pay, where the rate of pay is $35.00/hr for employees |

Your program must first find out which type of employee is being paid. You can use a menu with numeric choices for the user to enter:

**Enter type of employee:**

**1. Full-Time**

**2. Part-Time**

**3. Sessional**

**4. Partial Load**

After retrieving the employee type, your program must then calculate the amount to pay based on user inputs for that particular employee type.

Below are some sample interactions and output for this program:

*Sample Run for Employee Type 1*

**Enter Type of Employee**

**1. Full-Time**

**2. Part-Time**

**3. . Sessional**

**4. Partial Load**

**1**

**Enter Salary: 42500**

**Total Pay: $965.90**

*Sample Run for Employee Type 2*

**Enter Type of Employee:**

**1. Full-Time**

**2. Part-Time**

**3. Sessional**

**4. Partial Load**

**2**

**Enter Hours Worked: 40**

**Total Pay: $1200.00**

**And so on…**

*Sample run with invalid employee type*

**Enter Type of Employee:**

**1. Full-Time**

**2. Part-Time**

**3. Sessional**

**4. Partial Load**

**5**

**Error! You entered an invalid employee type.**

**Program-2**

An underground parking garage of a mall requires a program that allows clerks to calculate and display a bill for customer parking charges. A customer is charged when they leave the garage, and the amount charged is based on a few different factors. Some customers own a special membership card if they work in the building above the garage, so they are charged a special hourly rate of only $1 per hour. Customers without the card are charged $2 per hour, unless they park for more than 15 hours, in which case they are charged only $1.50 per hour. The Maximum charges allowed is $120, so if a customer's total charges are more than $120, they will be charged exactly $120.

Your program should ask the user whether or not the customer has a special membership card.

All literal numeric values in your program must be defined as constants. Be sure to use well-defined and appropriate names for all of your constants.

Here is an IPO chart to help you understand the program's requirements:

|  |  |
| --- | --- |
| **Inputs:** | - number of hours parked - whether or not the customer has a membership card |
| **Processing:** | - determine hourly rate based on membership status and hours parked - chargeable hours = ceiling of actual hours parked - total charges = hourly rate \* chargeable hours |
| **Outputs:** | - hourly rate - chargeable hours - total charges |

Your program should also include the following methods:

* Method Inputs:
  + hourly rate
  + chargeable hours
* The method returns a double value, the amount of the charges.

Output of your program should be formatted appropriately and should appear as follows

**Hourly Rate: $3.00**

**Number of Hours Charged: 6.0**

**Total Charge: $18.00**

**Don't forget to include proper documentation describing your code statements!.**

**Evaluation**

Your submission will be evaluated based on the following criteria:

**Efficient Code:** Program uses variables where and only when necessary; program doesn't define variables that are never used, nor does it use too many variables for unnecessary tasks; program logic is written concisely and is not cluttered with unnecessary tasks. Methods are written concisely and using the standards and techniques discussed in class.

**Functionality:** Program functions according to specifications.

**Programming Style:** Proper indentation and spacing, use of comments/documentation; all identifiers (variables, methods, constants, class) are descriptive and valid; variables are defined with appropriate types and converted when required. Method names follow the rules and standards discussed in class and have been given self-documenting names.