**Name : Jacob Nedumattathil Johar**

**ID: 102204195**

**Workshop 3**

**Date: 10 October 2020**

***Employee and Payable Case Study:***

***Logic***

***Payable***

- Declare the Payable class as interface with a method getPaymentAmount() with no implementation.

***Invoice***

* Declare the class Invoice which implements the class Payable as below

public class Invoice implements Payable

* The following data members are used

private String part;

private String Description;

private int count;

private double price;

* Implement the getter and setter methods for its data members

=> The price and count should be greater than 0 otherwise it should throw exception

throw new IllegalArgumentException(

"Price per item must be >= 0" );

* Implement the getPaymentAmount() as getQuantity() \* getPricePerItem()
* Implement the toString() method to return a string object.

***Employee***

* Declare the class Employee which implements the class Payable as below

public abstract class Employee implements Payable

* The following data members are used

private String first;

private String last;

private String ssn;

* Implement the getter and setter methods for its data members
* Implement the toString() method to return a string object.

***SalariedEmployee***

* Declare the class SalariedEmployee which extends the class Employee as below

public class SalariedEmployee extends Employee

* The following data members are used

**private** **double** weeklySalary;

* Implement the getter and setter methods for its data members
* Implement a parameterised constructor

public SalariedEmployee( String first, String last, String ssn,

double salary )

{

super( first, last, ssn ); // pass to Employee constructor

setWeeklySalary( salary );

}

* Implement the toString() method to return a string object.
* implement the getPaymentAmount() as below

**public** **double** getPaymentAmount()

{

**return** getWeeklySalary();

}

***HourlyEmployee***

* Declare the class HourlyEmployee which extends the class Employee as below

public abstract class HourlyEmployee extends Employee

* The following data members are used

private double wage;

**private** **double** hours;

* Implement the getter and setter methods for its data members

=> Wage should be greater than 0 otherwise throw exception

=> hours should be greater than 0 and less than 168. otherwise throw exception

* Implement a parameterised constructor
* public HourlyEmployee(String first, String last, String ssn, double Wage, double Hours) {
* super(first, last, ssn);
* setWage(Wage);
* setHours(Hours);
* }
* Implement the toString() method to return a string object.
* implement the getPaymentAmount() as below

**double** result = 0.0;

**if**(getHours() > 40 )

{

**double** value = getHours() - 40;

**double** wage = getWage() + (getWage() \* 1.5);

result = wage \* value + getWage()\* 40 ;

}**else**

{

result = getWage() \* getHours();

}

***CommissionEmployee***

* Declare the class CommissionEmployee which extends the class Employee as below

public abstract class CommissionEmployee extends Employee

* The following data members are used

**private** **double** grossSales;

**private** **double** commissionRate;

* Implement the getter and setter methods for its data members

=> grossSales should be greater than 0 otherwise throw exception

=> commissionRate should be greater than 0 and less than1. otherwise throw exception

* Implement a parameterised constructor
* public CommissionEmployee(String first, String last, String ssn, double sales, double rate) {
* super(first, last, ssn);
* setGrossSales(sales);
* setCommissionRate(rate);
* }
* Implement the toString() method to return a string object.

-implement the getPaymentAmount() as below

**public** **double** getPaymentAmount() {

**return** (getGrossSales() \* getCommissionRate())/100;

}

***BasePlusCommissionEmployee***

* Declare the class BasePlusCommissionEmployee which extends the class Employee as below

public abstract class BasePlusCommissionEmployee extends CommissionEmployee

* The following data members are used

**private** **double** baseSalary;

* Implement the getter and setter methods for its data members

=> baseSalary should be greater than 0 otherwise throw exception

public void setBaseSalary( double salary)

{

if ( salary > 0 )

baseSalary = salary;

else

throw new IllegalArgumentException( "Base Salary must be >= 0" );

}

* Implement a parameterised constructor
* public BasePlusCommissionEmployee(String first, String last, String ssn, double sales, double rate, double salary) {
* super(first, last, ssn, sales, rate);
* setBaseSalary(salary);
* }
* Implement the toString() method to return a string object.

- implement the getPaymentAmount() as below

public double getPaymentAmount() {

**double** value = (10 \* **super**.getPaymentAmount())/100;

**double** result = getBaseSalary() + value;

**return** result;

}

***PaymentEmployeeSystem***

* Define the main in this class
* => Create Payable object of size 4
* => To each index assign values of each subclass of Employee

payableObjects[0] = new CommissionEmployee("Jacob", "Johar", "437-213-888", 375.00, .5);

payableObjects[1] = new HourlyEmployee("Eldho", "Paul", "123-456-789", 100, 10);

payableObjects[2] = new SalariedEmployee("Aslam", "Karim", "324-678-964", 3000.00);

payableObjects[3] = new BasePlusCommissionEmployee("Deepak", "Hari", "778-543-555", 750.00, .5, 100.00);

=> A for loop for finding the specific class for each object

***Output***

