Date : 16th April 2022

Practical Assignment:

Aim :- To learn and implement the following concepts .

1) Classes and objects
2) Data members and member functions access
3) Constructors

Details of class, data members, member functions:

Class: Employee
Data members: Name Designation Basic salary Experience DA HRA IT GS NS Department

Member functions:
getData()
calc_DA()
calc_HRA()
calc_IT()
calc_GS()
calc_NS()
putData()
cal_Sal()

Designations	**B.S. (Basic salary)**	**O.D.A (% of Basic sal)**	**HRA(% of Basic sal)**
Manager	50,000	30	20
Supervisor	30,000	20	10
Executive	20,000	15	05

Gross_salary = Basic_salary + DA + HRA
Net_salary = GS - IT

DA to be calculated as (ODA + (Experience * 3)

IT to be calculated as:

30 %, if GS > 70000

20%, if GS > 50000

10%, if GS > 30000

0%, otherwise.

Basic:

1. Create Employee class and write main program to create its objects and do the following:.

Calculate DA, HRA, IT, Gross Salary and Net Salary.

Date : 23rd April 2022

Practical Assignment:

Aim :- To learn and implement the following concepts.

A) Array of Objects
B) Passing objects to the functions
C) Parameterized constructors.

Basic:

1. Create an Array of 10 objects of class named Employee and initialize them with default values as follows:

Name=yourname
Designation=”Emp”
Basic salary=0
Experience =0
DA=0
HRA=0
IT=0
GS=0
NS=0
Department=”Dept”

2. Take input of 10 employees for the above created array of objects and display them in following format:

Name	Desig	Dept	Exp	Basic	DA	HRA	IT	GS	NS
Aa	bb	cc	5	20000	6000	1000	2700	27000	24300

..
..
..

Moderate:

3. Count total number of specific type of employees.

4. Calculate total income tax paid by the organization using field IT.

5. Calculation of sum of salaries for various type of designations.

FYBSc. Div-B SET-1

Date : 29th April 2022

Practical Assignment:

Aim :- To learn and implement the following concepts.

A. Package
B. Simple Inheritance

<u>Basic:</u> (Simple Inheritance)

1. Create the class **Designation** as follows with given accessibility for its members**:**
NOTE: ("*" indicates private, "+" indicates protected, "++" indicates Public)

Class: Designation
Data members: +designation * basicsalary * oda_per * hra_per

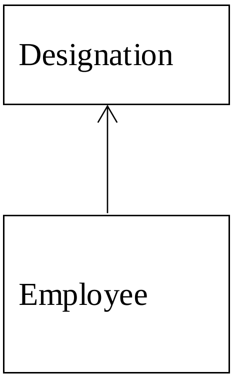
Deaignation(string des) (Constructor to initialize the object members based on designation passed to it)
++setDesig() (takes input of designation and assigns values to **basicsalary**, **oda** and **hra** automatically)
+desig() (returns **designation**, when called**)**
+basicSal() (returns **basicsalary**, when called**)**
+oda_per() (returns **oda %,** when called**)**
+hra_per() (returns **hra %,** when called**)**
++putDesigDetails() (Displays values of all data members of **Designation** class)

Write main program to test the class **Designation.**

2. *Make a copy of your previous assignment, and implement following additional functionality in the newly created copy:*

Inherit the class **Designation** in class **Employee** and Modify the accessibility of members of Employee class as follows: ("*" indicates private, "+" indicates protected, "++" indicates Public)

*<u>(OR you can create the new class **Employee** by inhering class **Designation** in it.)</u>*



Class: Employee **inherits class** Designation

Data members:
++Name
++Experience
*IT
*GS
*NS
*Department

Member functions:
++getData() (it should also take input of designation related details)
*calc_DA() **(** calculates and returns final actual **DA** value**,** when called**)**
*calc_HRA() **(** calculates and returns final actual **HRA** value**,** when called**)**
*calc_IT() **(** calculates and returns final actual **IT** value**,** when called**)**
*calc_GS() **(** calculates and returns final actual **GS** value**,** when called**)**
*calc_NS() **(** calculates and returns final actual **NS** value**,** when called**)**
++putData() (it should also display designation related details)
++cal_Sal() (calculates all components of salary and returns final salary)

Check proper working of your program and do necessary modifications , if required (only in your program without changing the above definition of class Employee).

<u>Moderate: </u>(multilevel inheritance)

1. Create following class by inheriting **Employee** class in it.

Class: ExecutiveTask

Data members:
*TaskName (name of the task assigned to employee)
* AssignedBy (name of the manager/supervisor who assigned the task)
*AssignDate (the date on which the task was assigned to employee)
*ExpCompDt (the expected completion date of the task)
*TaskStatus (“Working”/ “Suspended”/ “Resumed” / “Completed”)
*ActCompDt (the actual completion date of the task)
*DelayReason (Reason for the delay if any).

Member functions:
++assignTask() (a method to assign the task to an employee)
++showUpdateTaskStatus() (a method to show/update task status of an employee)
++compTask() (a method to close the task)

2. Initialize array of 10 Employee with values for 2 managers, 3 supervisors and 5 executives.
3. Write a program to assign task to “Executive” employees, show their status and complete the task.

<u>Input / Output :</u>

```
Select Operation:
============
1. Assign Task
2. Show Task Status
3. Close the Task

Enter your choice :  1

You are going to assign a new task to an executive.

Select the executive:
===============
     Name of the executive     Task status
1.   Ram                       Not Assigned
2.   Shyam                     Not Assigned
3.   Sita                      Not Assigned
4.   Gita                      Not Assigned
5.   Lakhan                    Not Assigned

Enter your choice :  3

You have selected to assign a new task to Sita.
Enter task details:
=============
```

TaskName : print assignment 5
AssignedBy (manager/supervisor) : Shankar
AssignDate : 29-Apr-2022
ExpCompDt :30-Apr-2022

Thank you..Shankar has assigned the task “Print assignment 5” to Sita.

Select Operation:
============

1. Assign Task
2. Show/update Task Status
3. Close the Task

Enter your choice : 2
You are going to display / update a task assigned to an executive.

Select the executive:

	Name of the executive	Task status
6.	Ram	Not Assigned
7.	Shyam	Not Assigned
8.	Sita	Assigned
9.	Gita	Not Assigned
10.	Lakhan	Not Assigned

Enter your choice : 3

You have selected to display / update status a task assigned to Sita.

Enter task details:
=============

TaskName : print assignment 5
AssignedBy (manager/supervisor) : Shankar
AssignDate : 28-Apr-2022
ExpCompDt :30-Apr-2022

Do you want to update the status(y/n): y

select New status of the task:
1. Working
2. Suspended
3. Resumed

Enter your choice : 1

You have selected to update status of a task assigned to Sita. The new status is “Working”. Thank You.

Select Operation:
============

4. Assign Task
5. Show/update Task Status

```
6.    Close the Task

Enter your choice :  3
You are going to close a task assigned to an executive.

Select the executive:
===============
      Name of the executive      Task status
1.        Ram                    Not Assigned
2.        Shyam                  Not Assigned
3.        Sita                   Working
4.        Gita                   Not Assigned
5.        Lakhan                 Not Assigned

Enter your choice :  3

You have selected to close a task assigned to Sita.

Enter task details:
=============
TaskName         : print assignment 5
AssignedBy  ( manager/supervisor  ) : Shankar
AssignDate      : 28-Apr-2022
ExpCompDt      :30-Apr-2022
Task Status       :Working
Enter Actual Compltion Date: 01-May-2020
Enter Reason for Delay : printer was not working on 29-Apr-2022.

The task status updated to :Complete

Thank you. The task assigned to Sita is closed now.

=========================================
```

Date : 30th April 2022

Practical Assignment:

Aim :- To learn and implement the following concepts:

A. Hierarchical Inheritance

***Note** : Make a copy of your previous assignment, and implement following additional functionality in the newly created copy:*

1. Create following class by inheriting **Employee** class in it.

Class: ManagerTask
Data members: *TaskName (name of the task assigned to employee) *AssignedBy (name of the manager/supervisor who assigned the task) * AssignedTo (name of the employee to whom the task is assigned) *TaskStaus (“**assigned**”/“Working”/ “Suspended”/ “Resumed” / “Completed”) (to be updated only by ReqTaskStatus() method)
Member functions: ++assignTask(Employee e) (a method to assign the task to an employee) ++ReqTaskStatus(Employee e) (a method to request task status of an employee) ++compTask(Employee e) (a method to close the task of an employee)

2. Write a program to assign task to “Executive” employees, show their status and complete the task.

FYBSc. Div-B SET-2

Date : 29th April 2022

Aim:- To learn and implement the following concepts.

A. Package

B. Simple Inheritance

Basic: **(Simple Inheritance)**

NOTE: Create MENU DRIVEN application

Write an OOP in java using the following information.

Create the class called **"Customer"** as follows with given accessibility for its members**:**

(NOTE: + indicates **Protected, *** indicates Public, ++ indicates **private)**

Class: Customer
Data member: + Name + Age + Contact

Note: Create a constructor to initialize the data members
Member functions:
* getCustomers()
* showCustomers()

1) Perform the following operations on it:

Input and display user details like name, age, contact of a customer.

Expected input:

```
Welcome! Enter following details:
Name: Peter Andrews
Age: 30
contact number: 4478962586
```

Expected Output:

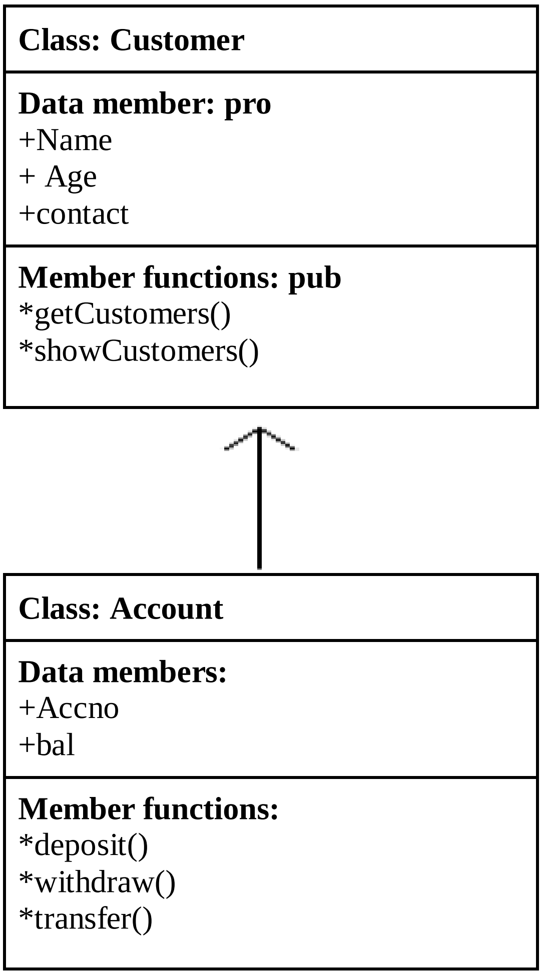
```
Name                 Age   Contact number
------------------------------------------
 Peter Andrews        30         4478962586
```

Moderate

2) Extend the above program to add the following functionalities

a) Create a class account “**Account** ” which will inherit the customer class.

(NOTE: + indicates **Protected, *** indicates Public, ++ indicates **private)**



Class: Customer

Data member: pro
+Name
+ Age
+contact

Member functions: pub
*getCustomers()
*showCustomers()

Class: Account

Data members:
+Accno
+bal

Member functions:
*deposit()
*withdraw()
*transfer()

Input and display user details like name, age, contact, accno, and bal for 3 customers

Expected Input:

Welcome! Enter following details:
Name: Peter Andrews
Age: 30
contact number: 4478962586
AccountNo: 111
Balance: 20000
.
.

(Repeat taking input for details of 3 customers)

•
•

Expected Output:

Name Age Contact number account number Balance

Peter Andrews 30 4478962586 111 20000

. ……………………………………..

(Show details of 3 customers)

………………………………………

Advanced

3) Allow the user to withdraw the balance from the account if the account balance is greater than 0 and if the withdrawal amount is less than the account balance. Display the final balance in the account after the transaction.

Expected Input:

Enter AccountNo: 111
Enter the amount you wish to withdraw: 30000

Expected Output:

Sorry! You don’t have a sufficient balance in your account.

Enter AccountNo: 111
Enter the amount you wish to withdraw: 10000
Transaction Successful!
Final balance: 10000

4) Deposit 10,000 in the account and show the reflected balance.

Expected Input:

Enter No: 111
Enter the amount you wish to deposit: 10000

Expected Output:

Transaction Successful!
Final balance: 20000