|  |
| --- |
| Hands-on Exercise Objective |
| After completing the hands-on exercises, you will be able to:   * Implement Inheritance in your program * Apply method overriding concept in inheritance |
| Scenario: In a company there are employees with two designations *Manager* and *Trainee*. Both employees share the same set of attributes and basic salary calculation logic is same but the basic salary of trainee and manager are different.The Manager has a travel allowance equal to 15% of the basic salary, whereas all the other employees the travel allowance is 10% of the basic salary. Write a program to maintain the entities using inheritance.Problem Statement 1:  1. Create a class Employee with the following instance variables.  |  |  | | --- | --- | | **Instance variables** | **Data type** | | employeeId | long | | employeeName | String | | employee Address | String | | employee Phone | Long | | basicSalary | double | | specialAllowance | double default value- 250.80 | | Hra | double,default value- 1000.50 |  1. Create an overloaded constructor in the employee class, which takes the below constructor parameters and initializes them to their respective instance variables.  |  |  | | --- | --- | | **Constructor parameter** | **Instance Variable** | | Id | employeeId | | Name | employeeName | | address | employeeAddress | | phone | employeePhone |  1. Create a method ***calculateSalary*** in which the basic salary needs to be calculated as below.   ***salary*** =  ***basicSalary*** + ( ***basicSalary*** \* ***specialAllowance***/100) + ( ***basicSalary*** \* **hra**/100);  The calculated salary should be displayed in the console.  **NOTE: salary is a local variable.**   1. Create the sub classes ***Manager*** and ***Trainee*** with base class ***Employee***. Create overloaded constructors which takes the below parameters and initializes them to their respective variables in the super class.  |  |  | | --- | --- | | **Constructor parameter** | **Instance Variable** | | id | employeeId | | Name | employeeName | | address | employeeAddress | | phone | employeePhone | | salary | basicSalary |  1. Create a class “***InheritanceActivity.java***” with a main method which performs the below functions,   **Test case #1:**   * Create an instance of ***Manager*** class by calling the overloaded constructor with the below parameters,  |  |  | | --- | --- | | **Constructor parameter** | **Instance Variable** | | id | 126534 | | Name | "Peter" | | address | "Chennai India” | | phone | 237844 | | salary | 65000 |  * Invoke the ***calculateSalary*** method of the manager object.   The salary calculated should be printed in the console.  **Test case #2:**   * Create an instance of ***Trainee*** class by calling the overloaded constructor with the below parameters,  |  |  | | --- | --- | | **Constructor parameter** | **Instance Variable** | | id | 29846 | | Name | "Jack" | | address | "Mumbai India” | | phone | 442085 | | salary | 45000 |  * Invoke the ***calculateSalary*** method of the trainee object.   The salary calculated should be printed in the console. Problem Statement 2:  1. Add a method called ***calculateTransportAllowance*** in ***Employee*** class which should calculate the transport allowance by calculating 10% (default allowance) of the salary. Print the salary after calculating.   ***transportAllowance*** = *10/100\*****basicSalary.***   1. For a manager, the transportation allowance is 15% of the basic salary. So override the ***calculateTransportAllowance*** method in ***Manager*** class which should calculate the transport allowance as 15% of the base salary. Print the salary after calculating.   ***transportAllowance*** = 15\****basicSalary*** /100.   1. For a trainee, the transport allowance is same as the default allowance; the method ***calculateTransportAllowance*** in the base class can be used. 2. Invoke the ***calculateTransportAllowance*** for the manager and trainee class in the main method of  ***InheritanceActivity.java.*** |

Q.2... Write a java class to develop an employee class object & initialized the emp name & emp Id for 5 Emp. now write another class having name and from this class you have to create 5 employee objects but you are not allowed to use new keyword.

Q.3... Develop a public java class and make sure nobody can create any object of this class from outside the class