SSN College of Engineering

Department of Computer Science and Engineering

UCS1313 – Object Oriented Programming Using Java Lab

II Year CSE - A Section (III Semester)

Academic Year 2019-20

Exercise – 2 - Inheritance

Sample Learning Outcome:

1. Need of inheritance and it's implementation in Java
2. Type of inheritance
3. Working of constructors in inherited class
4. Accessing inherited class through base class reference
5. Method overloading and overriding in inheritance

Best Practices:

1. Class Diagram usage
2. Naming convention – for file names, variables
3. Comment usage at proper places
4. Prompt messages during reading input and displaying output
5. Incremental program development
6. Modularity
7. All possible test cases in output

1. Write a java program with a class named 'Box' with following parameters name of the parcel, length, width and height and a function to calculate volume of box. Inherit a class named 'BoxWeight' from 'Box' with an additional member weight of the box in grams. Inherit 'BoxShipment' from 'BoxWeight' with an additional member shipmentcost per kilograms. Calculate the volume and cost for 'n' number of boxes on shipment and display in consolidated format. (Use Constructors)
2. Develop a java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 17% of BP as DA, 10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. The allowance for Programmer is Rs2000, Assistant Professor is Rs5000, Associate Professor is Rs10000 and Professor is Rs15000. Calculate the salary as grosssalary=BP+DA+HRA and deductions=PF+staffclubfund. Calculate the netsalary=grosssalary-deductions. Generate pay slips for the employees with their gross and net salary.

3. Write a java program with a class named ‘Person’ which consists of name, age, DOB and address. Have functions to get input and calculate_performance. Derive a class named ‘Student’ from ‘Person’ class with additional members like department, marks, extra-curricular. Calculate performance of Student (Outstanding, Excellent, Good, Fair) based on the grade and extra-curricular activities. Derive a class named ‘Professor’ from ‘Person’ with additional members like department, number of publications and funded projects. Calculate performance of Professor based on the number of publications and funded projects. In main get ‘n’ number of Person’s information and calculate the performance.