**Nilisha\_ inheritance\_assignment**

/\*1) Write a program to define finalize method for garbage collection. Display a message after garbage collection.

[Hint the finalize method is called when an object is about to get garbage collected. That can be at any time after

it has become eligible for garbage collection.]\*/

**package** lab4assignment;

**public** **class** GarbageCollector {

**static** **class** MyClass

{

**public** **void** finalize()

{

System.***out***.println("Garbage collection done");

}

}

**public** **static** **void** main(String[] args) {

MyClass c1 = **new** MyClass();

MyClass c2 = **new** MyClass();

c1 = **null**;

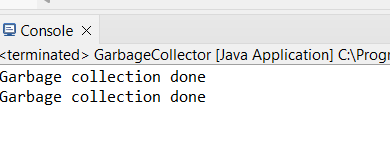
c2 = **null**;

System.*gc*();

}

}

**Output**

****

/\*2) Create an application for an organization having base abstract class named person containing abstract methods

\* like getData () and non abstract method named info.

a. Create an interface having password set to 123 and methods named displayData()

b. In class employee inherit both of class and interface and implement its behavior.

c. Employee class contains various fields like name, dept number and code.

d. Override getData and displayData () to retrieve and print employee record.

e. Create another class mainclass having main method, call above both methods in

mainclass.

f. Restrict mainclass from being inherited further.\*/

**package** lab4assignment;

**abstract** **class** Person {

**abstract** **void** getData();

**void** info() {

System.***out***.println("This is a non-abstract method.");

}

}

**interface** Authentication

{

String ***PASSWORD*** = "123";

**void** displayData();

}

**class** Employee **extends** Person **implements** Authentication

{

String name;

**int** deptNumber;

**int** code;

@Override

**void** getData()

{

name = "Nilisha";

deptNumber = 101;

code = 1001;

}

@Override

**public** **void** displayData()

{

System.***out***.println("Employee Name: " + name);

System.***out***.println("Department Number: " + deptNumber);

System.***out***.println("Employee Code: " + code);

}

}

**final** **class** main

{

**void** execute()

{

Employee employee = **new** Employee();

employee.getData();

employee.displayData();

employee.info();

}

}

**public** **class** Question2

{

**public** **static** **void** main(String[] args)

{

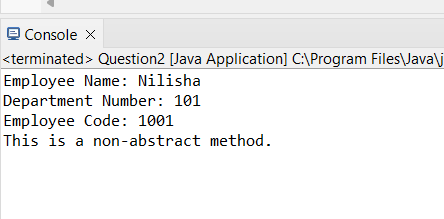
main m=**new** main();

m.execute();

}

}

Output



/\*3) Create three classes. named Order, ShippedOrder, and UseOrder.

The Order class will have four private instance variables. These variables will be for the customer name (String),

the customer number (integer), the order quantity (double) and the unit cost (double).

a. The Order class will have the following instance methods - get/set customer name, get/set customer number,

get/set order quantity, get/set unit cost. The Order class will also have a computePrice instance method.

This method should return the result of multiplying the order quantity times the unit cost.

b. The computePrice() method cannot have any arguments passed to it.

c. The ShippedOrder class will be a subclass of the Order class. This class will have one new private instance

variable.

d. This instance variable will store a shipping and handling charge. Set the shipping and handling charge to a

constant Rs.40.

e. The ShippedOrder class will have one new method. This method must be named computePrice and it

must override the computePrice method in the Order superclass. The computePrice method in the ShippedOrder class

should return the order cost (quantity times unit cost plus the shipping and handling charge).

No additional components should be displayed under these circumstances.

The UseOrder class will display the customer name, customer number, order quantity, and unit cost.

After the appropriate object is created, print all the values of the instance variables and the order's

total cost.

f. You must use the get methods to retrieve the values from the object and you must use the computePrice method to

"retrieve" the order's total cost. \*/

**package** lab4assignment;

**class** Order {

**private** String customerName;

**private** **int** customerNumber;

**private** **double** orderQuantity;

**private** **double** unitCost;

**public** String getCustomerName() {

**return** customerName;

}

**public** **void** setCustomerName(String customerName) {

**this**.customerName = customerName;

}

**public** **int** getCustomerNumber() {

**return** customerNumber;

}

**public** **void** setCustomerNumber(**int** customerNumber) {

**this**.customerNumber = customerNumber;

}

**public** **double** getOrderQuantity() {

**return** orderQuantity;

}

**public** **void** setOrderQuantity(**double** orderQuantity) {

**this**.orderQuantity = orderQuantity;

}

**public** **double** getUnitCost() {

**return** unitCost;

}

**public** **void** setUnitCost(**double** unitCost) {

**this**.unitCost = unitCost;

}

**double** computePrice() {

**return** orderQuantity \* unitCost;

}

}

**class** ShippingOrder **extends** Order {

**private** **final** **int** charge = 40;

@Override

**double** computePrice() {

**return** **super**.computePrice() + charge;

}

}

**public** **class** UseOrder {

**public** **static** **void** main(String[] args) {

ShippingOrder order = **new** ShippingOrder();

order.setCustomerName("Nilisha");

order.setCustomerNumber(998844);

order.setOrderQuantity(21);

order.setUnitCost(10);

System.***out***.println("Customer Name: " + order.getCustomerName());

System.***out***.println("Customer Number: " + order.getCustomerNumber());

System.***out***.println("Order Quantity: " + order.getOrderQuantity());

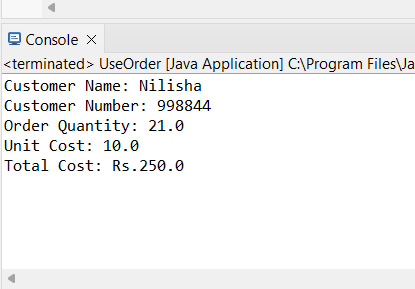
System.***out***.println("Unit Cost: " + order.getUnitCost());

System.***out***.println("Total Cost: Rs." + order.computePrice());

}

}

Output:



/\*4) Create a class phonebook having fields like name ,phone ,address and also contains nested class personal having field

like relation and method like getinput and putinput() .Take all necessary input for record .

a. Create another nested static class business having fields like organization, dept, mobile . and methods like accept

and show

b. Create another class containing main function , that ask employee choice

c. If choice of user is one ask personal record. Else take business entry and display them\*/

**package** lab4assignment;

**import** java.util.Scanner;

**public** **class** PhoneBook

{

String Name;

String Address;

**int** phoneNo;

**class** Personal

{

**void** getinput()

{

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Enter Name: ");

Name = s.nextLine();

System.***out***.println("Enter Address: ");

Address = s.nextLine();

System.***out***.println("Enter Phone Number: ");

phoneNo = s.nextInt();

s.close();

}

**void** putinput()

{

System.***out***.println(" Name: " + Name);

System.***out***.println("Address: " + Address);

System.***out***.println("Phone no: " + phoneNo);

}

}

**static** **class** Business

{

String Org;

String Dept;

**int** mobileNo;

**void** accept()

{

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Enter Organization Name: ");

Org= s.nextLine();

System.***out***.println("Enter Department Name: ");

Dept = s.nextLine();

System.***out***.println("Enter Mobile Number: ");

mobileNo = s.nextInt();

s.close();

}

**void** show()

{

System.***out***.println("Organization Name: " + Org);

System.***out***.println("Department Name: " + Dept);

System.***out***.println("Mobile no: " + mobileNo);

}

}

**public** **static** **void** main(String[] args) {

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Enter your choice:");

System.***out***.println("1. Personal Record");

System.***out***.println("2. Business Record");

**int** c = s.nextInt();

**if** (c == 1)

{

Personal P = **new** PhoneBook().**new** Personal();

P.getinput();

P.putinput();

}

**else** **if** (c == 2)

{

Business B = **new** Business();

B.accept();

B.show();

}

**else**

{

System.***out***.println("Invalid choice");

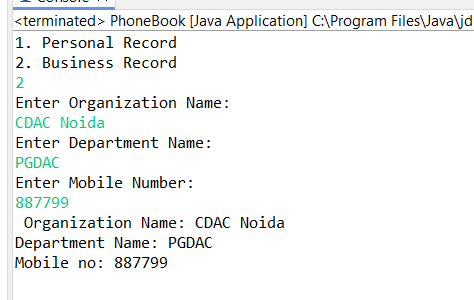
}

s.close();

}

}

Output:



/\*5. Write a program in java. A class Teacher contains two fields Name and Qualification.

\* Extends the class to department it contains dept. no and Dept Name. An interface named as college it

\* contains one field name of the college. Using the above classes and interface get the appropriate information

\* and display it.\*/

**package** lab4assignment;

**class** Teacher

{

String name;

String qualification;

**public** Teacher(String name, String qualification)

{

**this**.name = name;

**this**.qualification = qualification;

}

**public** **void** displayTeacherInfo()

{

System.***out***.println("Teacher Name: " + name);

System.***out***.println("Teacher Qualification: " + qualification);

}

}

**class** Department **extends** Teacher **implements** College

{

**int** deptNo;

String deptName;

**public** Department(String name, String qualification, **int** deptNo, String deptName)

{

**super**(name, qualification);

**this**.deptNo = deptNo;

**this**.deptName = deptName;

}

**public** **void** displayDepartmentInfo()

{

System.***out***.println("Department No: " + deptNo);

System.***out***.println("Department Name: " + deptName);

}

}

**interface** College

{

String ***collegeName*** = "IGNOU";

}

**public** **class** CMS {

**public** **static** **void** main(String[] args)

{

Department dept = **new** Department("Archana Suman", "Ph.D", 101, "Computer Science");

System.***out***.println("College Name: " + College.***collegeName***);

dept.displayTeacherInfo();

dept.displayDepartmentInfo();

}

}

Output:

